

# トランジスタ技術

## SPECIAL

特集 画像処理回路技術のすべて  
カメラとビデオ回路, パソコンとを融合させる

No.5



CQ出版社



好評発売中

# トランジスタ技術 SPECIAL No.22

## 特集 デジタル回路ノイズ対策技術のすべて

TTL/CMOS/ECLの活用法と誤動作/トラブルへの処方



B5判 160頁 定価 1,540円 送料 260円

数年前のMPUのスピードというところ4 MHzもあればすごく速いという部類だったのですが、最近では10 MHz, 16 MHz, 25 MHz, 33 MHzとどんどん高速化してきています。

こうなってくるとデジタル回路といえども高周波回路的な発想がないと、回路を正しく安定に働かせることができません。また、これらの高速回路が発生するEMIノイズに対してもいろいろな規制がしかれるようになってきました。

そこでこの号では、高速化されてきた最近のデジタルICとその応用回路、さらにEMIノイズ対策技術を、実験写真を多く示しながらやさしく紹介します。

■Contents■ 第1部 デジタル回路のノイズ対策 実務編  
第2部 デジタルIC回路の設計 実務編

### 既刊 (各B5判 2色刷) 定価1,540円 送料260円)

- |  |  |
|--|--|
| No.1 個別半導体素子活用法のすべて<br>基礎からマスタするダイオード、トランジスタ、FETの実用回路技術      | No.11 フロッピー・ディスク・インターフェースのすべて<br>需要の急増するFDDシステムの基礎から応用           |
| No.2 作りながら学ぶMC68000<br>16ビットMPUとその周辺LSIを使いこなすためのハード&ソフト      | No.12 入門ハードウェア 手作り測定器のすすめ<br>電子回路設計の基礎と実践へのアプローチ                 |
| No.3 PC9801と拡張インターフェースのすべて<br>16ビット・パソコンを使いこなすためのハード&ソフト     | No.13 シミュレータによる電子回路理論入門<br>コンピュータを使ったアナログ回路設計の手法を理解するために         |
| No.4 C-MOS標準ロジックIC活用マニュアル<br>実験で学ぶ4000B/4500B/74HCファミリ       | No.14 技術者のためのCプログラミング入門<br>MS-C, Quick C, Turbo Cによるソフトウェア設計のすべて |
| No.5 画像処理回路技術のすべて<br>カメラとビデオ回路、パソコンと融合させる                    | No.15 アナログ回路技術の基礎と応用<br>計測回路技術のグレードアップをめざして                      |
| No.6 Z80ソフト&ハードのすべて<br>基礎からマクロ命令を使いこなすまでのノウハウを集大成            | No.16 A-D/D-A変換回路技術のすべて<br>アナログとデジタルを結ぶ最新回路設計ノウハウ                |
| No.7 HD64180徹底活用マニュアル<br>Z80を越えた高性能8ビットCPUのすべて               | No.17 OPアンプによる回路設計入門<br>アナログ回路の誤動作とトラブルの原因を解く                    |
| No.8 データ通信技術のすべて<br>シリアル・インターフェースの基礎からモデムの設計法まで              | No.18 ホビー・エレクトロニクス入門<br>電子回路を作る楽しさと実現する喜びを求めて                    |
| No.9 パソコン周辺機器インターフェース詳解<br>セントロニクス/RS-232C/GPIB/SCSIを理解するために | No.19 PC9801計測インターフェースのすべて<br>オリジナル拡張ボードでパソコンを実践活用しよう            |
| No.10 IBM PC & 80286のすべて<br>世界の標準パソコンとマルチタスクの基礎を理解する         | No.20 アナログ回路シミュレータ活用術<br>ゲーム感覚の回路設計を体験しよう                        |
|  | No.21 デジタル・オーディオ技術の基礎と応用<br>最もポピュラーな最新技術を理解しよう                   |

年間購読をご希望の方は、年間購読料10,500円(送料、税込)を現金書留または郵便振替(東京0-10665)で、CQ出版(株)経理部までお申し込みください。バックナンバーは最寄の書店から注文できます。弊社へ直接ご注文の場合は、定価、送料を添えてCQ出版(株)営業部宛へお願いいたします。

CQ出版社

〒170 東京都豊島区巣鴨1-14-2 ☎03-947-6311 振替東京0-10665

(定価は税込です)





〈写真1〉  
ビデオ・スルー  
画面



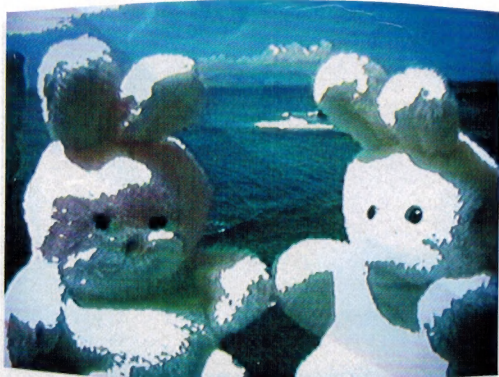
〈写真2〉  
白文字のスーパー  
インポーズ



〈写真3〉  
赤文字のスーパー  
インポーズ



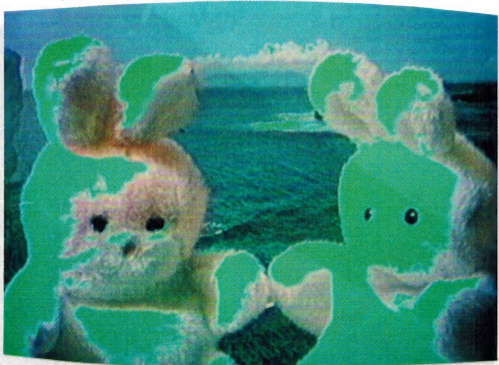
〈写真4〉  
バック・グラウ  
ンドを青にして  
白文字をスーパー  
インポーズ



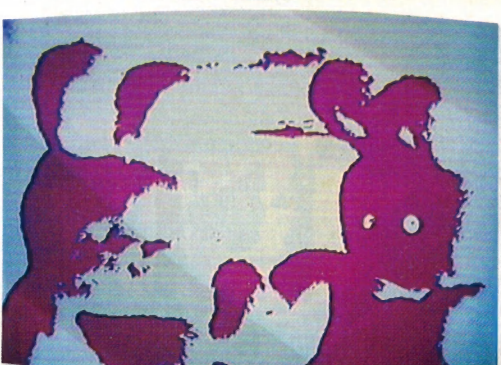
〈写真5〉  
輝度整形を行っ  
た画面



〈写真6〉  
輝度整形+着色  
(Y)

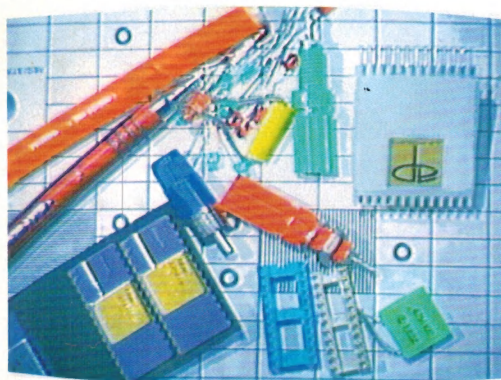


〈写真7〉  
輝度整形+着色  
(G)

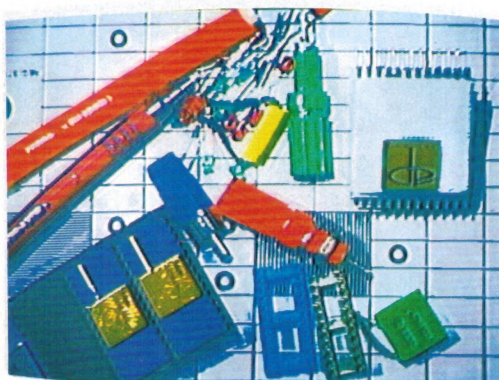


〈写真8〉  
輝度整形+着色  
(M)+背景反転  
(白)





〈写真1〉  
原画像(1)



〈写真2〉  
写真1のディフ  
ェクト



〈写真3〉  
原画像(2)



〈写真4〉  
写真3のソラリ  
ゼーション



〈写真5〉  
白黒カメラによ  
る原画像



〈写真6〉  
写真5の2値化  
画像



〈写真7〉  
疑似カラー(a)



〈写真8〉  
疑似カラー(b)

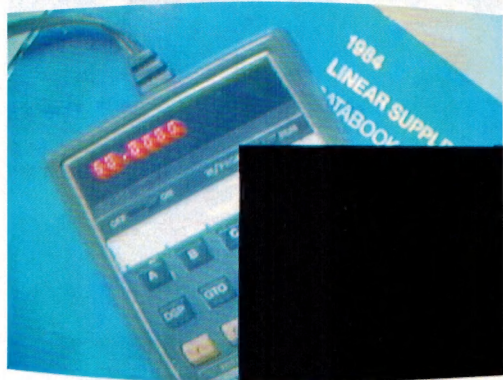




〈写真9〉  
フェード前



〈写真10〉  
フェード後



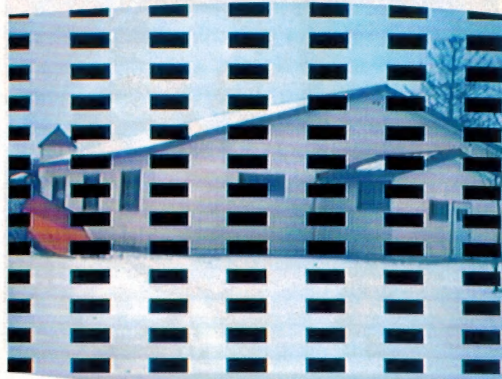
〈写真11〉  
ワイプ(a)



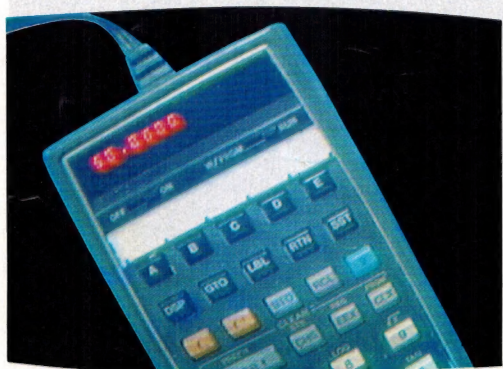
〈写真12〉  
ワイプ(b)



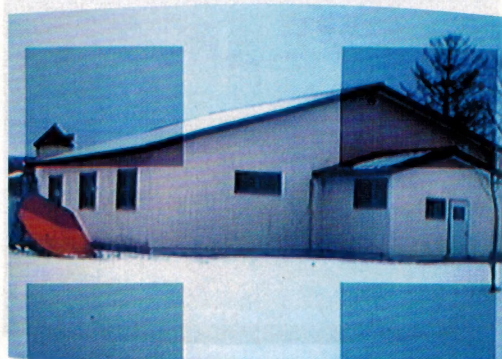
〈写真13〉  
スリット(a)



〈写真14〉  
スリット(b)



〈写真15〉  
スライス



〈写真16〉  
レベル・シフト





〈写真17〉  
モザイク(a)



〈写真18〉  
モザイク(b)

## カラー口絵

## Chapter11

## パソコン画像処理での映像例



〈写真1〉  
疑似カラー(a)



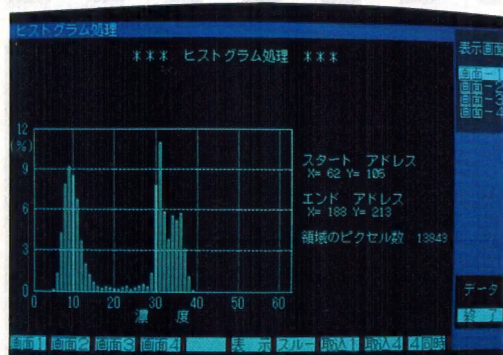
〈写真2〉  
疑似カラー(b)



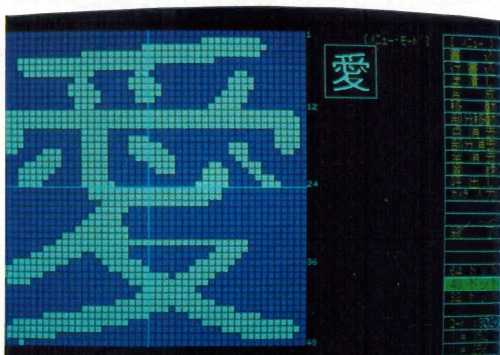
〈写真3〉  
疑似カラー(c)



〈写真4〉  
疑似カラー(d)



〈写真5〉  
ヒストグラム  
処理



〈写真6〉  
文字フォント作  
成への利用



# トランジスタ技術 SPECIAL

No.5

CONTENTS

## FEATURES

カメラとビデオ回路、パソコンとを融合させる

## 画像処理回路技術のすべて

- 映像信号の基本であるNTSCコンポジット信号と、そのY/C分離技術など
- Chapter 1 ● NTSC信号とその利用技術 ● 松井俊也 ..... 2
- NTSCコンポジット信号をRGB信号に分割する。CCDくし形フィルタを使用
- Chapter 2 ● NTSCデコーダの設計・製作 ● 角田和宏/長谷川孝美 ..... 11
- RGB信号をNTSCコンポジット信号に変換する。RFモジュレータも紹介
- Chapter 3 ● NTSCエンコーダの設計・製作 ● 角田和宏/渡辺成治 ..... 26
- 画像をハード/ソフトにする機能。NJM2209を使用
- Chapter 4 ● ビデオ・エンハンサの設計・製作 ● 藤崎 功/峯岸英雄 ..... 36
- スーパーインポーズの一般論と、RGBインターフェースによるスーパーインポーズ方式を紹介
- Chapter 5 ● NTSCデコーダ&スーパーインポーズの設計・製作 ● 宏 祐二/松井俊也 ..... 41
- コンポジット信号の中にキャラクタ・ジェネレータの文字をカラーでスーパーする
- Chapter 6 ● カラー・スーパーインポーズの設計・製作 ● 千葉雅彦/峯岸英雄 ..... 51
- 専用IC ZNA234を使用
- Appendix ● ビデオ・パターン・ジェネレータの製作 ● 南雲敏行 ..... 63
- ビデオ用A-Dコンバータの基本的な応用技法を紹介
- Chapter 7 ● NTSC信号のA-D変換技術 ● 尾津加 河湖 ..... 66
- 疑似カラー、ディフェクト、ワイプ、フェータ、スリット、モザイクなどの画像効果を作る
- Chapter 8 ● 多機能ビデオ・エフェクタの設計・製作 ● 尾津加 河湖 ..... 71
- シリアル・アクセス・メモリ μPD41221Cを使用したデジタル・ステル・メモリ
- Chapter 9 ● ビデオ・フィールド・メモリの設計・製作 ● 岩崎 潔 ..... 83
- ライン・メモリ μPD41101Cを使用したインタレース→ノンインタレース変換
- Chapter 10 ● 倍速スキャン・コンバータの設計・製作 ● 湯川達雄 ..... 97
- 市販のPC98用画像入力ボードを使用した画像処理システムの実験
- Chapter 11 ● パソコンによる画像処理技術 ● 桑村伸一/坂巻 顯 ..... 106
- PC98用で、比較的容易に実現可能なイメージ・ボードの紹介
- Chapter 12 ● パソコン用画像入力ボードの設計・製作 ● 小野隆夫 ..... 118
- PC98用の工業用ライン・センサ・カメラ・インターフェースの紹介
- Chapter 13 ● パソコン用1次元イメージ・センサ・ボードの設計・製作 ● 白川光英 ..... 137
- 入手できる2次元CCD ICX018CKIによるカメラ製作の紹介
- Chapter 14 ● 2次元CCDカメラの設計・製作 ● 長谷川孝美 ..... 147
- PC98用のデジタル・カメラと、その応用技術を紹介
- Chapter 15 ● パソコン用2次元デジタル・カメラの製作と応用 ● 寺本三省 ..... 164







## FEATURES

最近ではテレビ関連の技術者以外にも、NTSC映像信号を扱う例が増えています。すなわち、テレビやテレビ・カメラ、VTR、レーザ・ディスクなどとコンピュータとの融合です。本特集では、この分野の技術をくまなく、具体的な設計・製作例として紹介しています。

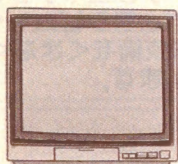
カメラとビデオ回路, パソコンとを融合させる

# 画像処理回路技術のすべて





## NTSC信号とその利用技術



松井俊也

画像信号の基礎となっているのはNTSC規格による映像コンポジット信号です。ここでは、このNTSC信号の性質をつかみ、輝度信号と色信号との分離方法について紹介します。

最近、画像信号を使った電子機器が多く開発され商品化されてきました。例えば、テレビやVTRにおいても、従来までのただ見るだけのものから、画像信号を駆使していろいろと遊びの要素を取り入れたものが登場してきています。また、そのなかで高画質化、高品質化への動きが活発化してきています。

ここでは、まず基本にたち戻り、NTSCの画像信号はどのような信号なのかを説明し、次にこの信号の基本的な扱い方および高画質化へのアプローチについて、実際の回路例をもとに説明していきます。

### NTSC信号とは

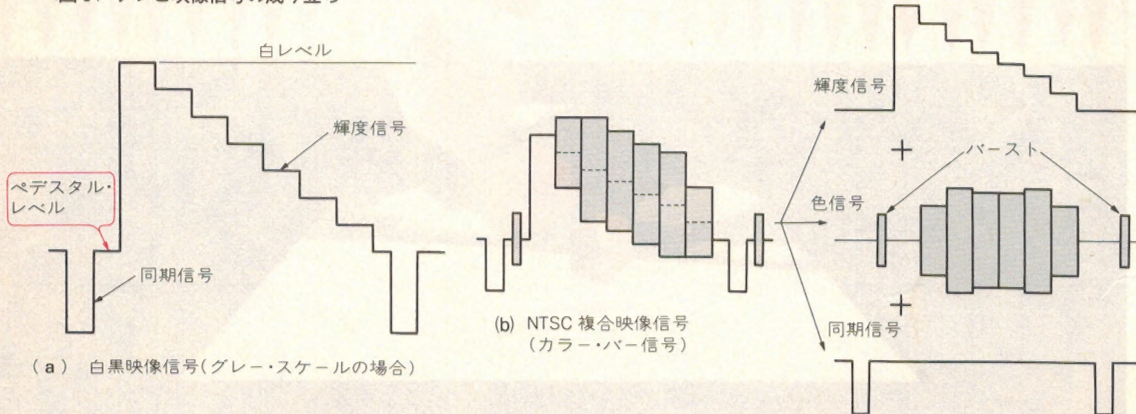
NTSC信号とは、**National Television System Committee**で決定された信号であり、たいへん巧妙に作りあげられた信号です。

このNTSC信号の大きな特徴のひとつに、**カラー信号と白黒信号との両立性**があげられます。この信号は、いったいどのような構成になっているのでしょうか。

#### ● NTSC信号の内わけ

NTSC信号は図1に示すように大きく分類して、**明暗を表現する輝度信号と色を表現する色信号**、そして画面の位置およびタイミングを伝える情報としての**同期信号**に分けられます。

〈図1〉テレビ映像信号の成り立ち



この三種類の信号は、**お互いに影響を与えないよう**に合成されているのが一つの特徴です。NTSC信号を周波数スペクトルで表すと、図2のようになっています。輝度信号の周波数帯域幅4.2MHzの内側に色信号3.58MHzが挿入されています。

たんにこの図だけで判断すると、色信号が輝度信号に大きな影響を与えるように見えますが、実際には、色信号の周波数と水平同期信号の周波数の関係は、図3のように**輝度信号のエネルギー・スペクトラムの間に、色信号のエネルギーを挿入**するようにしています(周波数インターリービングという)。この関係は次のようになっています。

$$f_{sc} = \frac{455}{2} \times f_H \dots \dots \dots (1)$$

$$= 3.579545 \text{ MHz}$$

$f_{sc}$ : 色副搬送波周波数

$f_H$ : 水平同期周波数

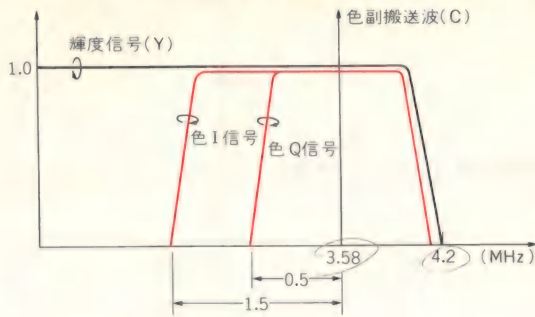
NTSC信号の輝度信号は、図3のように $f_H$ の高調波の周波数にピーク・エネルギーをもつスペクトラムとなり、輝度信号と色信号の関係は図4のように表せます。

また、NTSC信号には、水平、色信号のほかに垂直信号、音声中間周波信号も重畳されます。それぞれの周波数関係を(2)式～(4)式に示します。

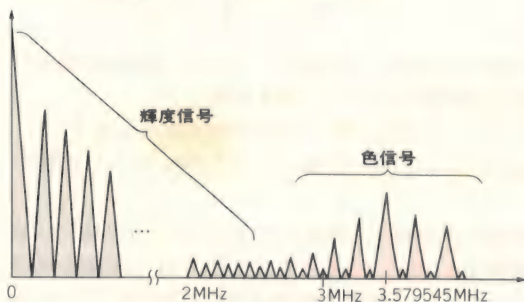
$$\text{音声中間周波数 } f_{SIF} = 4.5 \text{ MHz} \dots \dots \dots (2)$$



〈図2〉 NTSC信号周波数スペクトラム



〈図4〉 NTSC Y/C信号スペクトラム



$$\text{水平同期周波数 } f_H = \frac{4.5}{286} \dots\dots\dots(3)$$

$$\approx 15.734\text{kHz}$$

$$\text{垂直同期周波数 } f_V = \frac{2 \times f_H}{525} \dots\dots\dots(4)$$

$$\approx 59.94\text{Hz}$$

### ● NTSCの走査線

では、NTSC信号での各ライン(走査線)間の関係はどのようなになっているのでしょうか。

まず、走査のはじめの1ライン目の信号を(5)式とします。各ライン番号は図5を参照してください。

$$E_1 = E_Y + E_I \sin \omega_c t + E_Q \cos \omega_c t \dots\dots\dots(5)$$

ここで、

$E_1$  : 1ライン目の映像コンポジット信号

$E_Y$  : 輝度信号電圧

$E_I$  : I信号電圧 } 色信号

$E_Q$  : Q信号電圧

$\omega_c$  : 色信号周波数 (=  $2\pi f_{sc}$ )

$$E_2 = E_Y + E_I \sin \omega_c (t + t_H) + E_Q \cos \omega_c (t + t_H)$$

$$E_3 = E_Y + E_I \sin \omega_c (t + 2t_H) +$$

$$E_Q \cos \omega_c (t + 2t_H)$$

⋮

$$E_n = E_Y + E_I \sin \omega_c (t + nt_H) + E_Q \cos \omega_c (t + nt_H)$$

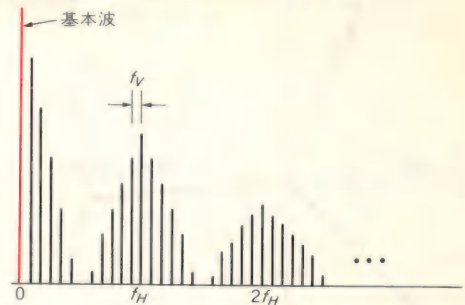
$$t_H = 1/f_H$$

上式をわかりやすく変形すると、

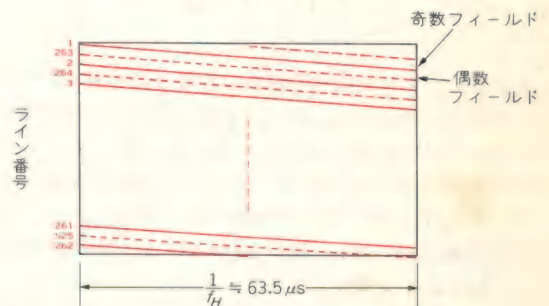
$$E_n = E_Y + E_I (-\sin \omega_c t) + E_Q (-\cos \omega_c t) \dots\dots\dots(6)$$

$$(n = 1, 3, 5, \dots)$$

〈図3〉 NTSC輝度信号スペクトラム



〈図5〉 NTSC走査線



〈図6〉 RGB信号による色の再現(デジタルRGBの場合)

G	R	B	色
0	0	0	黒
0	0	1	青
0	1	0	赤
0	1	1	紫
1	0	0	緑
1	0	1	水色
1	1	0	黄色
1	1	1	白

1 : ON  
0 : OFF

$$E_n = E_Y + E_I \sin \omega_c t + E_Q \cos \omega_c t \dots\dots\dots(7)$$

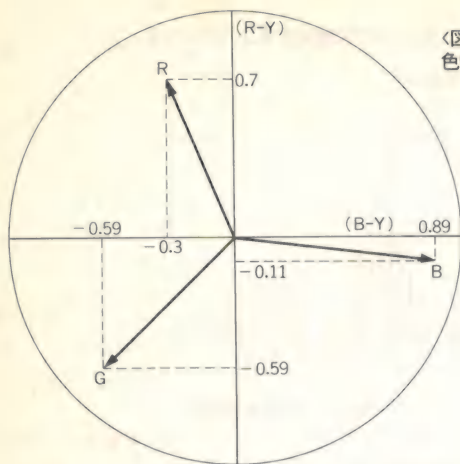
$$(n = 0, 2, 4, \dots)$$

この関係は、あとで説明するくし形Y/C分離(輝度信号/色信号分離)の項で使用しますので、詳しくは後にゆずることにします。

### ● NTSCの色信号

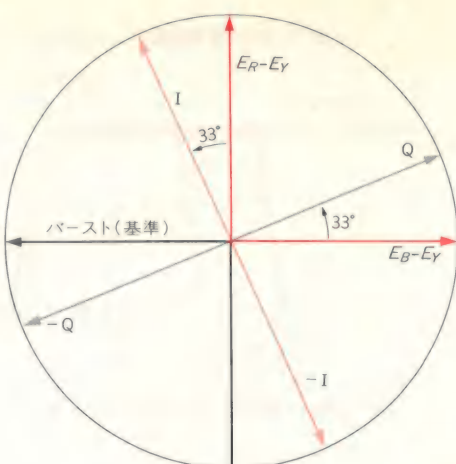
色には一般によく知られている光の3原色R(赤), G(緑), B(青)がありますが、テレビもまた、このR・G・B信号を利用して色を表現しています。この様子を図6に示します。ここでは簡単のためにR・G・B信号を「あるか、ないか」のデジタルで示しているため、7色の表現ですが、実際のテレビなどでは各R・G・B信号がアナログ的ですので、ほぼ無限に近いカラー表示が得られるわけです。





〈図7〉  
色信号スペクトラム

〈図8〉  
NTSC色信号の基本ベクトル図



ただし、テレビ信号の場合の色信号は白黒テレビとの共用という点から、**R・G・B原色信号から輝度成分Yを差し引いた(R-Y), (G-Y), (B-Y)といった色差信号を使用しています。**そして、実際の受信では(R-Y), (B-Y)のみを色情報として復調し、(G-Y)は受信側で(R-Y)と(B-Y)をベクトル合成することで得るような構成になっています。

では、R・G・B信号と輝度信号Yや色差信号との関係はどのようになっているのでしょうか。これは人間の視感度から、各信号は、

$$\begin{aligned} E_Y &= 0.3E_R + 0.59E_G + 0.11E_B \\ E_{R-Y} &= 0.7E_R - 0.59E_G - 0.11E_B \\ E_{G-Y} &= -0.3E_R + 0.41E_G - 0.11E_B \\ E_{B-Y} &= -0.3E_R - 0.59E_G + 0.89E_B \end{aligned}$$

となっており、それぞれある一定の割合で合成されています。図7にR・G・B信号のベクトルを示します。

また、人間が一番細かいところ(周波数の高い信号)まで見ることができる色は、オレンジあるいは肌色であるため、この信号の周波数帯域は1.5MHzまで、またこの軸と90°位相のずれた軸の信号帯域は、0.5

MHzまでに制限して送信しています。前者を**I信号**、後者を**Q信号**と呼びます(図2参照)。

これらの信号のベクトル図を図8に示します。I, Q軸は(R-Y), (B-Y)軸から33°位相シフトした軸です。

実際の色復調は、現在では主に(R-Y), (B-Y)軸で復調をし、(G-Y)を合成して作り出す方式を採用しており、I, Q軸での復調はあまり行われていません。

## 色信号、輝度信号の分離方法

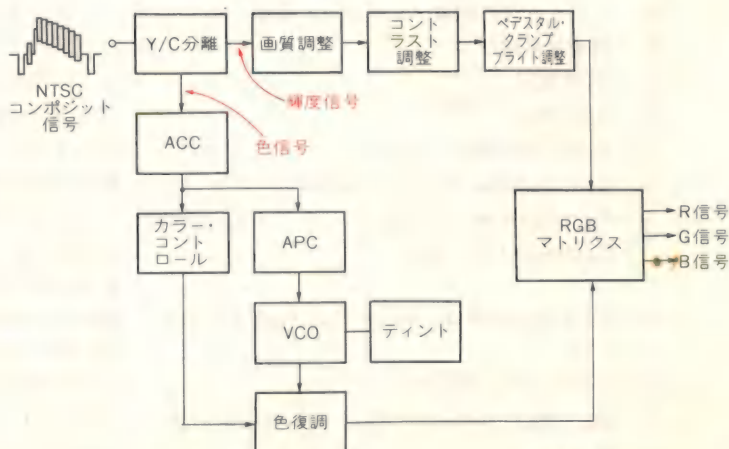
NTSC信号について前項で説明しましたが、実際にはどのようにしてそれぞれの信号を分離し、R・G・Bのコンポーネント信号を作り出しているのでしょうか。

ここでは、まずその前段階であるNTSC信号から、色信号と輝度信号を分離する方法について紹介します。

### ● 画質を低下させないための信号分離の考え方

NTSCコンポジット信号からコンポーネント信号を作り出す処理について、図9のブロック図をもとに、

〈図9〉コンポジット信号処理ブロック図





説明を行います。

まず、輝度信号と色信号との分離を行うわけですが、このブロックで、輝度信号の特性(解像度)が決定されます。後でもう少し詳しく述べますが、NTSCシステムの場合、色信号が3.58MHzと映像信号帯域の4.2MHzの内側にあるため、色信号が輝度信号に影響を与えないように輝度信号と色信号との分離(Y/C分離)を行う場合、たんなる周波数分離(トラップやバンドパス・フィルタ)では色信号の周波数特性が悪くなってしまう。

また実際には、伝送路中で周波数特性が劣化したり、輪郭を強調したい場合には、水平方向の解像度を変える画質補正回路および、輝度信号の振幅を調整して、輝度に応じたコントラストを設定するコントラスト・コントロール回路が必要となります。

さらに、輝度信号のAC成分のみを伝送していくと、(容量結合により)輝度信号の大きさによって輝度信号のペダスタル、黒および白の信号電圧が変化してしまいます。

例えば、白の少ない信号の時には白が画面でつぶれたようになり、白の多い信号の時には灰色になったりしてしまいます。そこで、DC再生という形で輝度信号のペダスタル部をクランプし、DCの伝送を行うようにしています。

このために必要なのがペダスタル・クランプ回路です。

### ● 色信号処理の考え方

色信号処理には、Y/C分離で得られた色信号を使用しますが、この色信号の処理には以下の回路が必要となります。

## 映像信号の規格とレベル

テレビの色信号には、色相および彩度(色のあざやかさ)の情報が含まれます。色相は3.58MHzの基準バーストに対する色信号の位相で表され、彩度はその振幅で表されます。この関係を図Aに示します。

輝度信号は明るさを表す信号で、直流成分が明るさを決定します。

コンポジット信号の振幅は、75Ωで終端した場合1V<sub>P-P</sub>と規定されており、

同期分 0.285V<sub>P-P</sub>

輝度分 0.715V<sub>P-P</sub>

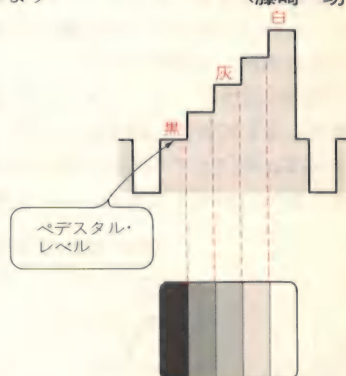
と定められています。

同期信号と輝度信号の境界がビデオ信号の基準レベルとなりますので、このレベルをペダスタル・レベルと呼びます。

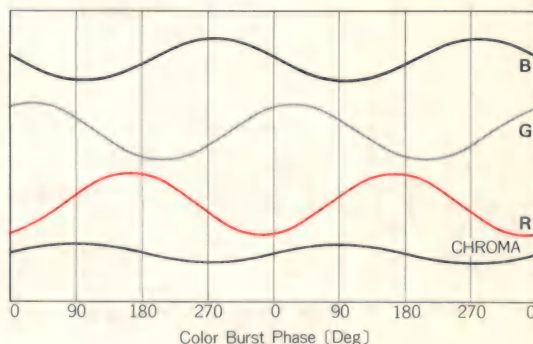
図Bに示すように、輝度信号0Vすなわちペダスタル・レベルでは黒、0.715Vが白を表します。そして、この間で順次、黒→こい灰色→うすい灰色→白のように変化します

〈藤崎 功〉

〈図B〉  
輝度信号の波形と  
画面の対応



〈図A〉 色相の分布





まず、入力された色信号の復調はすべてバースト信号を基準に行われるため、復調段までレベルおよび位相の忠実な伝達が必要となります。そのため、バーストのレベル(色信号)が変化したとき、R・G・Bの出力電圧が変化しないようにACC(Auto Color Control)増幅を行います。

そしてこのACCブロックでバーストのレベルを検出し、色信号の大きさをバーストの大きさに追従するようにしています。また、カラー・コントロールとして、色信号の大きさを変化させ、RGBの出力振幅を変化させます。

さらに、バーストの周波数および位相に同期した色信号復調キャリアを作るためのVCO(電圧制御発振器)、APC(Auto Phase Control)回路が必要となります。この復調キャリアを使い、(R-Y)、(B-Y)、I、Q信号を復調します。(G-Y)は、(R-Y)と(B-Y)のベクトル合成で再生し、それぞれの信号に輝度信号Yを加算してR・G・B信号を再生します。

### ● 従来のフィルタによるY/C分離

NTSC信号の説明で述べたように、色信号が輝度信号帯域内に位置していることから、輝度信号と色信号の分離の手法によって、輝度信号の特性が決定するといっても過言ではありません。このY/C分離を従来の手法とくし形フィルタによる分離の手法について、実際の回路を使用して説明します。

まず図10に従来のハイパス・フィルタおよびトラップを使用したY/C分離回路を示します。図10において $R_1$ と $R_2$ は $Tr_1$ のバイアス抵抗で動作点の設定を行っています。また、エミッタ抵抗、コレクタ抵抗の定数を

同じにして、利得がないようにしてあります。

$Tr_1$ のコレクタ抵抗 $R_3$ は、トラップおよびディレイ・ラインのインピーダンス・マッチング抵抗も兼ねています。また $C_2$ と $C_4$ 、 $L_1$ でハイパス・フィルタを構成しています。

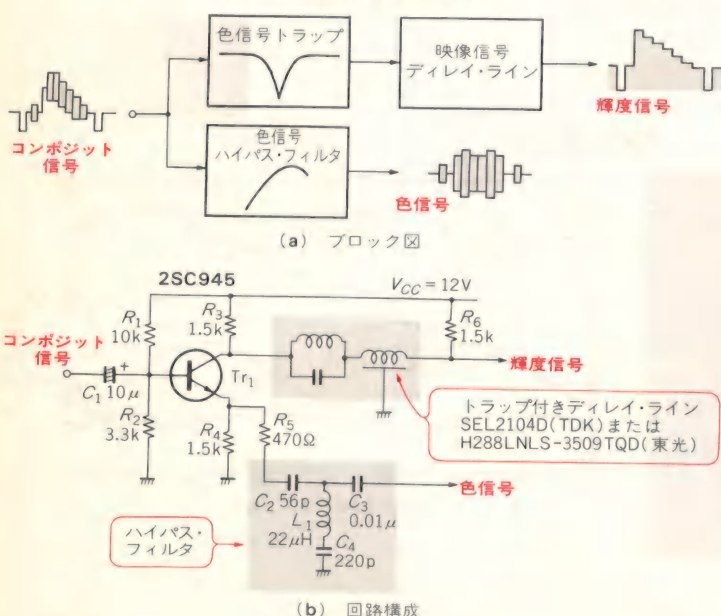
このフィルタの特性を写真1に示します。写真2はトラップ付きディレイ・ライン〔H288LNLS-3509 TQD, 東光(株)〕の周波数特性です。3.579545MHzにトラップが設定されており、そのため映像周波数特性も3MHz弱のポイントで減衰がはじまっています。

このフィルタの特性は、中間周波増幅段の入力IFフィルタの特性に合わせる必要があります。というのは図11に示すように、入力IFフィルタ特性により、色信号の帯域特性がフラットではなくなるためです。その補正をしてトータルでフラットにするため、ハイパス・フィルタを用います。

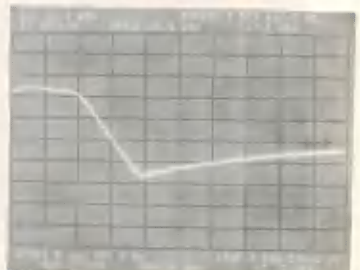
また、輝度信号側のディレイ・ラインは、色信号のハイパス・フィルタでのディレイ時間とACC増幅、色信号復調回路で発生する時間差を補正するためのもので、およそ数百nsのものです。通常のIC(例えば輝度信号処理IC  $\mu$ PC1352Cなど)で約100ns程度のディレイが考えられます。

写真3に図10の回路によるY/C分離の特性を示します。この中で左側の写真が、NTSCコンポジット信号です。この信号を図10の回路でY/Cの分離を行うと、右側の上下の写真となります。写真の中で2段に分かれているうちの上段部を拡大(時間軸)したものが下段の信号です。ライン(水平走査線)間での相関を理解しやすくするために、垂直帰線期間と走査期間の分かれ

〈図10〉従来のY/C分離回路



〈写真1〉色信号分離ハイパス・フィルタの周波数特性



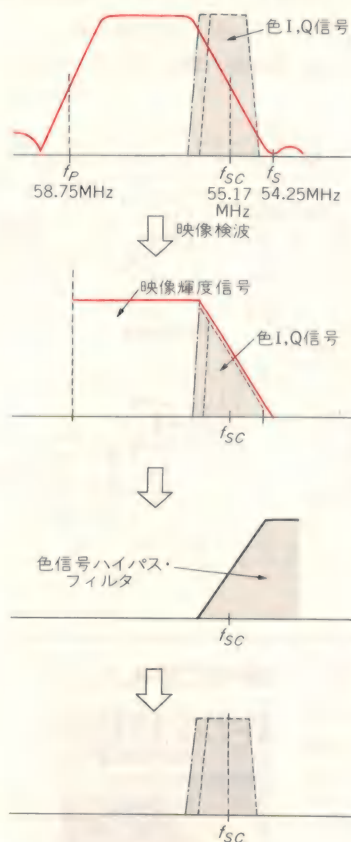
〈写真2〉輝度信号ディレイ・ラインおよびトラップの周波数特性



るタイミングでのY/C分離の様子を示しています。また、右上の写真が色信号で、右下の信号が映像信号です(レベルが反転しており、上が同期側となっている)。

### ● くし形フィルタの原理

次にY/Cくし形フィルタについて図12を用いて説明



〈図11〉  
色信号帯域特性  
の補正

します。まず基本原理について述べます。

くし形フィルタは次のことが前提条件としてあげられます。

- (1) 色信号の位相が1 Hごとに反転していること (NTSCインターリービング)。
- (2) 1 H前後の信号の類似性。

以上の条件を基にその動作を説明します。輝度信号は、1 H(1水平走査期間)の時間差をもつ信号を互に加算して得ることができます。つまり前述の(6)式、(7)式より、例えば2ライン目と3ライン目を加算すると、

$$E_2 + E_3 = E_{Y2} + E_{Y3} + (E_{I2} - E_{I3}) \sin \omega_c t + (E_{Q2} - E_{Q3}) \cos \omega_c t$$

ここで、 $E_{Y2} = E_{Y3}$ ,  $E_{I2} = E_{I3}$ ,  $E_{Q2} = E_{Q3}$  とすると、

$$E_2 + E_3 = 2 E_{Y2} = 2 E_{Y3}$$

となります。また、色信号は、1 Hの時間差をもつ信号を互いに減算することにより得ることができます。

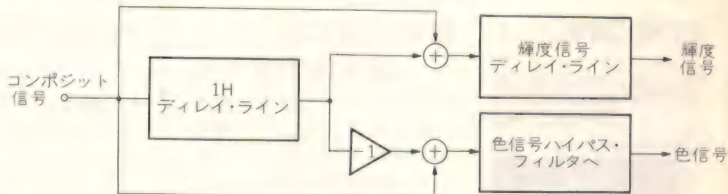
つまり、同様に(6)式、(7)式を用いて表すと、

$$E_2 - E_3 = E_{Y2} - E_{Y3} + (E_{I2} + E_{I3}) \sin \omega_c t + (E_{Q2} + E_{Q3}) \cos \omega_c t$$

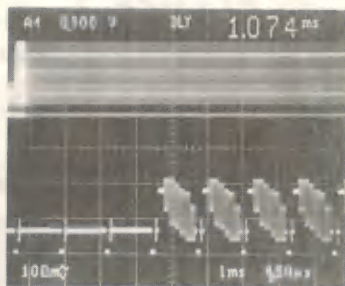
ここで  $E_{Y2} = E_{Y3}$ ,  $E_{I2} = E_{I3}$ ,  $E_{Q2} = E_{Q3}$  とすると、

$$E_2 - E_3 = 2 E_{I2} \sin \omega_c t + 2 E_{Q2} \cos \omega_c t$$

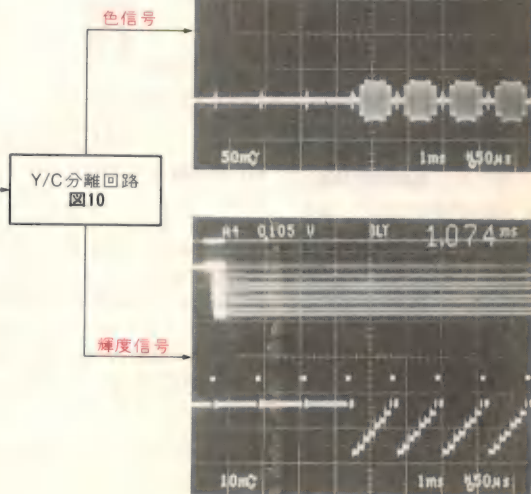
〈図12〉 くし形フィルタの原理



〈写真3〉  
従来フィルタ回路  
によるY/C分離



NTSCコンポジット信号



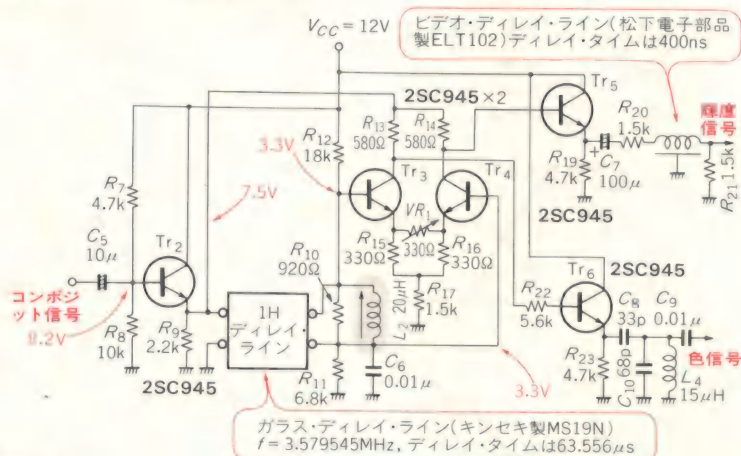


となります。

### ● くし形フィルタの回路

実際の動作を図13を用いて説明します。 $R_7$ と $R_8$ はエミッタ・フォロワのDC動作点を設定しており、約8.2Vにバイアスしています。これは、 $Tr_3$ と $Tr_4$ にその直流分を使用するため、高めに設定しています。 $Tr_2$ のエミッタ・フォロワでインピーダンス変換されたY/Cの信号は、1Hのディレイ・ラインに入力され、1H遅延され、 $Tr_3$ のベースに入力されます。

〈図13〉 くし形フィルタ回路例



定価:すべて1,540円

### SPECIAL No.1



### 個別半導体素子 活用法のすべて

一般用/電源整流用/定電圧/定電流/発光/可変容量/マイクロ波の各ダイオード、小信号・汎用/パワー/小信号高周波/高周波高出力の各トランジスタ、小信号・汎用FET/パワーMOS FET/高周波用FET

既刊

### PC9801と拡張インター フェースのすべて

PC9801シリーズの内部構成、PC9801シリーズの拡張スロットの詳細、メモリ・ボードの製作、A-DボードとD-Aボードの製作、メカトロニクス・ボードの製作、他

・・・好評発売中

### SPECIAL No.3



### SPECIAL No.2



### 作りながら学ぶ MC68000

MC68000とCPUボードの製作、モニタ・プログラムの搭載、命令セットとアセンブラ文法、I/Oボードの製作、FDCボードの製作、FORTHの移植手順、FORTHの使い方とツール、RAMボードの製作、CP/M-68Kの移植(1)(2)(3)、C言語とシステムのグレードアップ

### C-MOS標準ロジック IC活用マニュアル

基本ゲートIC、インターフェース用IC、ラッチIC、フリップフロップ、マルチバイブレータ、カウンタ、デコーダ/エンコーダ、シフトレジスタ、ディバイダ、ディスプレイ・ドライバ、データ・セクタ、デジタル・コンパレータ、レジスタ・ファイル、パリティ・チェック用IC

### SPECIAL No.4





また色信号は、 $Tr_6$ でインピーダンス変換され、 $C_8$ 、 $C_{10}$ 、 $L_4$ で構成されるハイパス・フィルタを通過後出力されます。

図13の回路では、1Hのディレイ・ラインとしてガラスのディレイ・ライン(超音波ディレイ・ライン)を使用しています。通過帯域にも制限があり、約 $\pm 0.8$  MHzです。

写真4が図13のくし形フィルタを用いたY/C分離の様子を示したものです。左上の写真が図13中の $Tr_3$ のコレクタ信号波形であり、右上が $C_9$ の出力信号波形です。また、左下は $Tr_4$ のコレクタ信号波形です。左

下の写真で1H前の信号とまったく相関のないラインでは、Y/C分離がなされず、色信号が残っているのがよくわかると思います。

写真5がくし形フィルタの周波数特性です。写真(a)は、図13中の $Tr_4$ のコレクタ出力の周波数特性を示したもので、 $15.734\text{kHz}(=f_H)$ の周波数の整数倍で、くしの歯のようにトラップを形成しています。写真(b)は $3.579545\text{MHz}$ 付近を拡大(時間軸)したものです。

図14に、従来のフィルタとくし形フィルタとの周波数特性の違いを示します。

また、写真5にくし形フィルタの周波数特性を示し

## NTSC画像信号用のディレイ・ラインとフィルタ

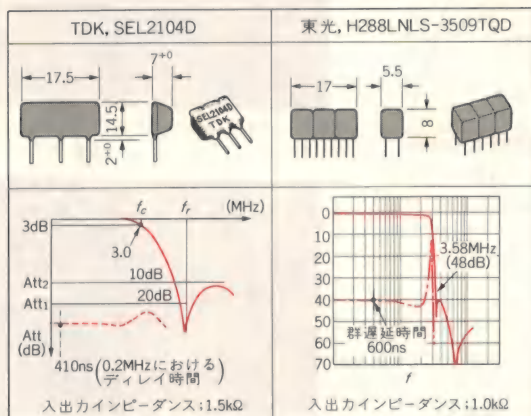
NTSCのコンポジット信号を扱うとき、フィルタとディレイ・ラインが重要な働きをするのは本文で述べたとおりです。

しかし、これらの部品は特性的に、また経済的に、一般のLCR個別部品で構成することはかなり大変です。したがって一般的には、専門メーカの手でモ

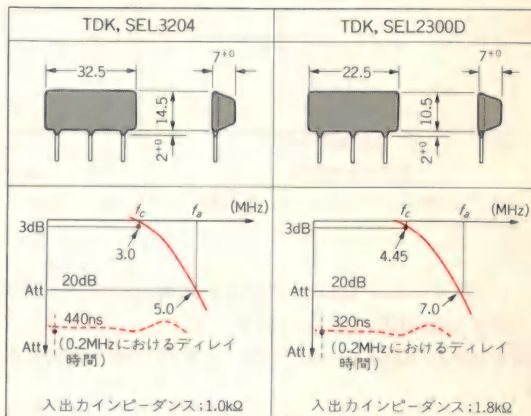
ジュール化され、標準品として発売されています。

図Cに、この章のY/C分離回路で使用しているディレイ・ライン、ローパス・フィルタの例を示します。(a)と(b)はLCの複合部品でアセンブルしたものの、(c)はガラスを媒体とすることにより温度特性を向上させたものです。

〈図C〉ビデオ用ディレイ・ラインとフィルタ



(a) トラップ付きディレイ・ライン、ローパス・フィルタの例



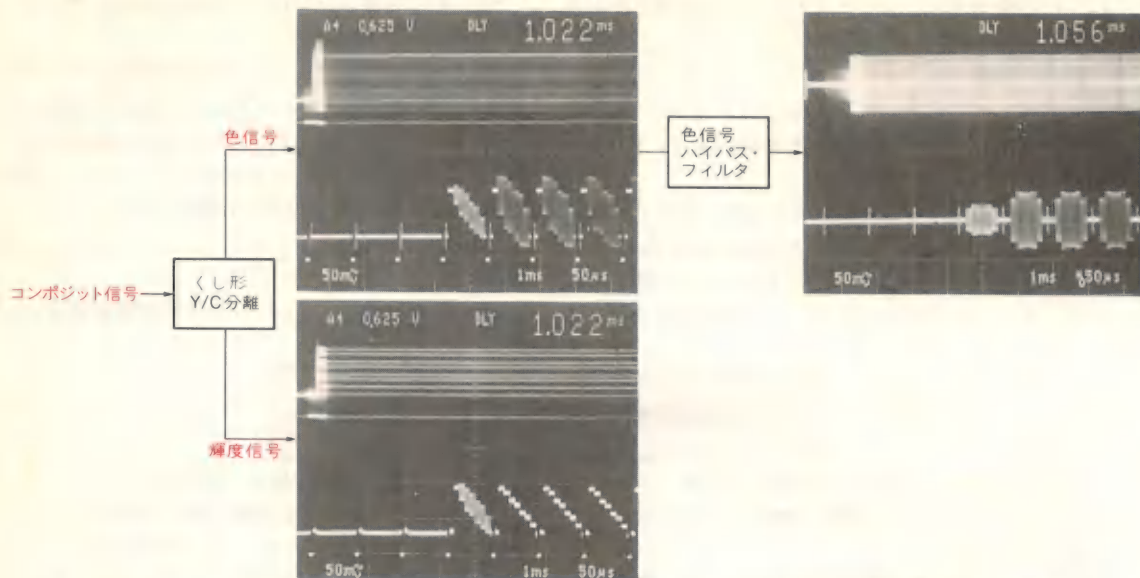
(b) ディレイ・ライン、ローパス・フィルタの例

公称周波数	$f_0$	3.579545	MHz
遅延時間	$\tau_D$	63.556	$\mu\text{s}$
遅延時間偏差	$\Delta\tau_D$	$\pm 5$	ns (max)
挿入損失	$I \cdot L$	$10 \pm 3$	dB
通過帯域幅	$BW$	$\pm 0.75$	MHz (min)
不要反射 (スプリアス)	3次 (3 $\tau$ )	30	dB (min)
	その他 ( $\tau_n$ )	30	dB (min)
遅延時間温度特性 (0~60°C)		$\pm 5$ (20~50°C)	ns (max)
動作可能温度範囲		-20~+60	°C
終端抵抗	$R_t$	560	$\Omega$
整合インダクタンス	$L_t$	15	$\mu\text{H}$

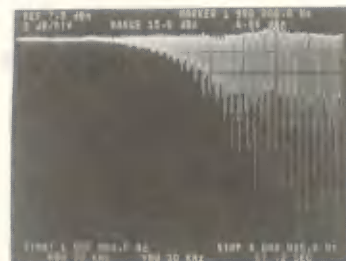
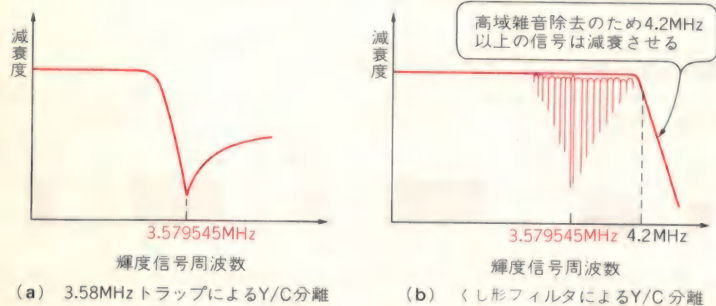
(c) ガラス(超音波)ディレイ・ライン (キンセキ(株)MS19N)



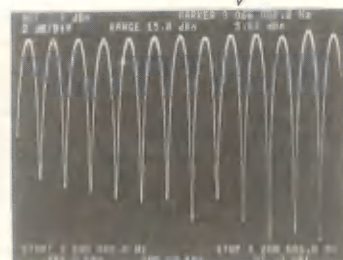
〈写真4〉くし形フィルタによるY/C分離



〈図14〉トラップおよびくし形フィルタによるY/C分離の周波数特性



(a) TrAのコレクタ



ます。

くし形フィルタには、このほかに最近になって登場した**CCDを使用**したものがあります。これについては次の章のNTSCデコードのところで実例を紹介します。

#### 参考文献

- (1) 日本放送協会編：NHKカラーテレビ受信技術，日本放送出版協会。

〈写真5〉  
くし形フィルタの  
周波数特性



## NTSCデコーダの設計・製作



角田和宏  
長谷川孝美

画像処理に利用しやすい映像信号は、R・G・Bに分離された信号です。ここではCCDによるくし形フィルタを使用したY/C分離回路をもつ、コンポジット→RGBコンバータについて紹介します。

最近のAV(オーディオ・ビジュアル)機器の発達は目ざましいものがあり、高解像度テレビや高画質VTRが簡単に手に入るようになりました。それに伴い、一昔前までは放送局や放送機器メカの独壇場であった画像処理も、一部はそれらの機器に取り入れられて、私たちでも手軽に楽しむことができるようになりました。

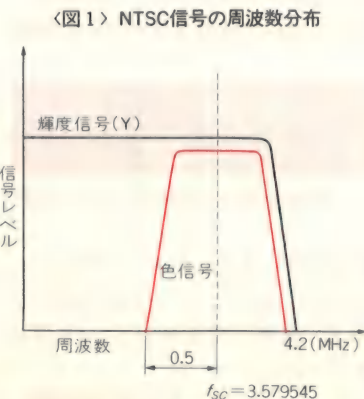
しかし、もっと多くの機能が欲しいと思っても、価格競争の激しいメーカ品では、なかなかこちらが思っているような余計な機能は付けてくれません。それならば自分で作ってしまえ、と思ってもちょっととつづくにくいのがビデオの世界です。

そこで、ここでは画像処理の基礎ともいえるべきコンポジット映像(NTSC)信号とRGB信号とを扱ううえでの、ツールとして欠かすことのできない**NTSC→RGB信号コンバータ(NTSCデコーダという)**についての設計・製作法を紹介することにします。

### NTSCエンコーダ/デコーダの必要性

#### ● コンポジット映像信号の欠点

最近の**高性能テレビ**には、ステレオ入出力端子やVTR専用端子と共に、必ずといっていいくらい**RGB入力端子**が付いています。これは、現在のテレビ方式で、可能な限り画質を向上させようとした場合の一つの結果なのです。



**コンポジット映像信号**では、第1章で述べた通り、**輝度、色、同期の三つの成分**が、帯域4.2MHzの中にひしめき合って混在しているといえます。

したがって、この中から三つの成分を抜き出すのは至難のワザです。例えば、図1のような信号から輝度信号と色信号を分離する場合(これを**Y/C分離**という)、図2の(a)のような特性をもつフィルタを通せば良いのですが、実際このフィルタは図(b)のように、カットオフ周波数がオーバーラップしてしまい、残って欲しい成分まで消え去ってしまいます。

通常、どちらかといえば**色の再現性が重視**されるので、結果的には輝度信号の周波数の高い成分が削られ、その分、**解像度が落ち**るのです。

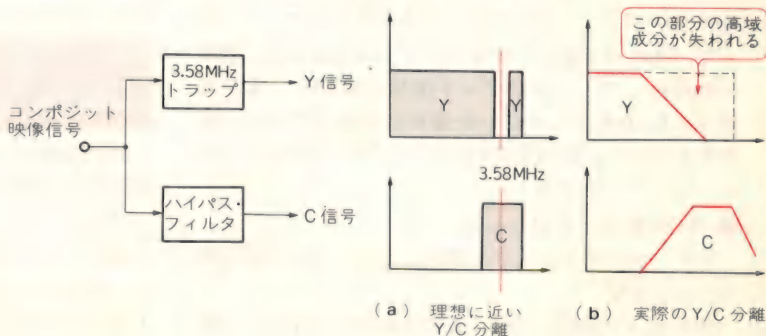
最近では、この点を改善するために**くし形フィルタ**というものが各メーカでしのぎをはずって開発されていますが、それでも無キズのまま三つの成分を分離することはできません。

そこで、伝送するには少しめんどですが、各成分をはじめから分離して送れば様々な点で有利になります。これがRGB方式と呼ばれるものです。

#### ● RGB方式の数々のメリット

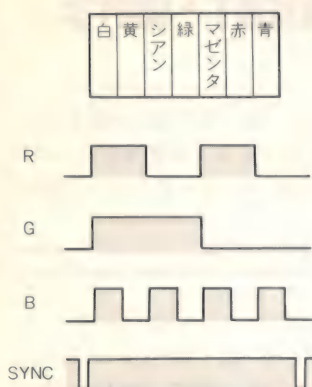
RGB方式では、図3に示すような**R・G・Bの原色信号と、同期信号をRGB21ピン・マルチコネクタ**(48ページのコラム参照)を通して、それぞれ別個の線で接続します。なにしろ初めからY/C分離もせず、マトリクス回路を通す必要もないので、解像度が高く、画

〈図2〉 フィルタとトラップによるY/C分離の特性

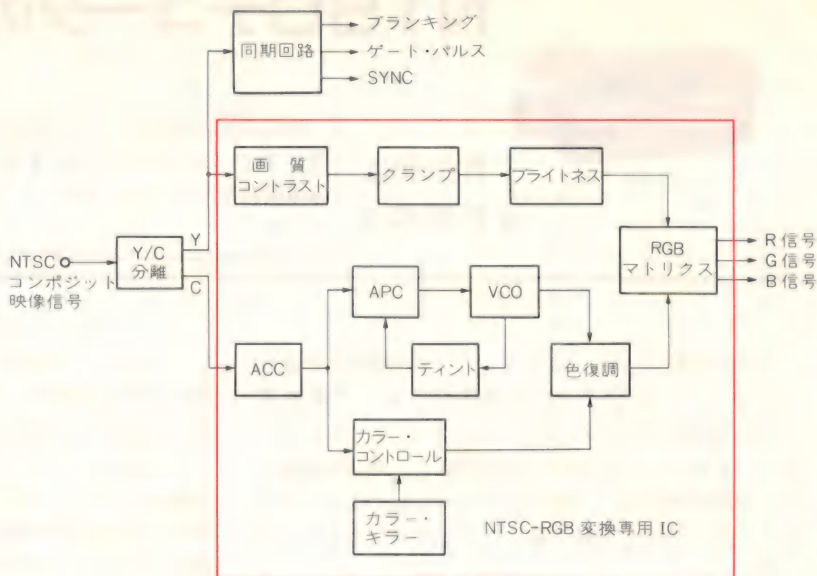




〈図3〉RGB方式



〈図4〉NTSCデコーダ・ブロック図



質劣化の少ない画像が期待できます。

また、画像処理を考えた場合もRGB方式のほうが処理しやすいです。例えば、色バランスを変えたい場合、コンポジット信号では前述の通り、色はバーストと色信号の位相差によって決まります。したがって、色信号の全体をバーストに対してズラすことは可能ですが、あくまでも相対的に色がズレたにすぎません。

かといって、ある特定の色だけを変えたいからといって、その位相に相当する部分だけを抽出するなどということは、まず無理でしょう。

その点、RGB方式だと、R・G・Bが独立に操作できますから、それぞれのレベルを変えてやるだけで任意の色に作りかえることができます。

コンピュータとのインターフェースも、RGBでやりとりすることはご承知の通りです。

RGB方式の欠点として、接続がめんどうであるとか、映像信号と同期信号がズレる場合があるという問題がありますが、このように画質の面を考えれば、コンポジット信号よりもRGB信号に軍配が上がります。

一方、手軽に入手でき、しかも安い映像機器はまだまだコンポジット信号のものが多くあります。したがって、それらを使って画像処理をする場合には、前述の理由により、コンポジット信号をRGBに、また処理後にRGBをコンポジット信号に変換してやる必要があるのです。前者をNTSCデコーダ、後者をNTSCエンコーダと呼びます。

#### ● NTSCデコーダのしくみ

まず、NTSCデコーダの基本となるブロック図を図4に示します。

順を追って説明します。入力されたコンポジット映

像信号は、Y/C分離回路によって輝度信号と色信号に分けられます。輝度信号は画質(シャープネス)やコントラスト、それにクランプされた後でブライトネスの調整を受け、マトリクス回路に入ります。

一方の色信号は、最初に大切なバースト信号の振幅変動を制御して、常にレベルを一定に保つACC(Automatic Color Control)回路に入ります。

APC(Automatic Phase Control)回路では、バーストとVCO(Voltage Controlled Oscillator)の発振出力を比較して、VCOの発振動作を制御します。VCOでは色の復調に必要な、90°位相差をもった二つの基準位相信号を作ります。

また、ティント・コントロール回路は、色副搬送波とバースト間の位相を操作することにより、色相の調整を行います。

一方、カラー・コントロール回路では、色信号の振幅を操作して色の濃度を調整します。また、白黒信号が入ったとき、色信号回路が誤動作して不快な色雑音が出ないように、バースト信号を検出して、白黒信号入力時は色信号が出ないようにするカラー・キラー回路がついています。

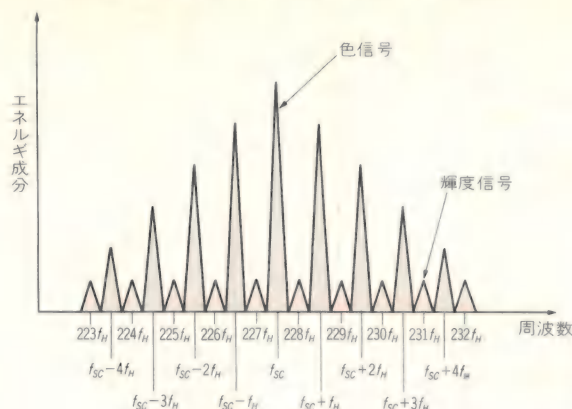
### Y/C分離回路 —— くし形フィルタの設計 ——

画像処理において、解像度を決定する最も大きな要因がY/C分離です。

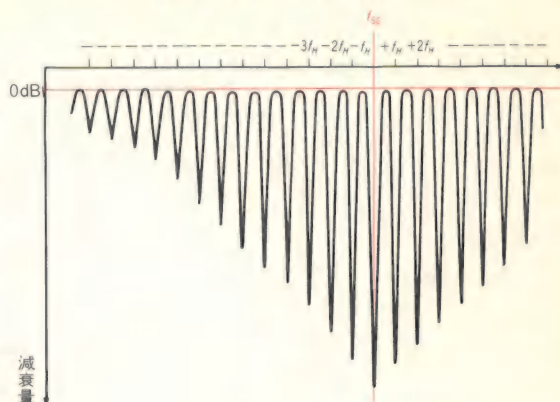
方式としては、第1章でも述べたトラップ方式とくし形フィルタ方式とがありますが、ここでは今流行のくし形フィルタしかも、最新のICを使用したCCDくし形フィルタを考えてみます。



〈図5〉周波数インターリーピング( $f_{sc}$ 近傍)



〈図6〉くし形フィルタの特性



### ● くし形フィルタの原理

NTSC信号は、図5に示すようなエネルギー・スペクトラムをもっていますから、そこから輝度信号が欲しければ、色信号のある部分だけを抜き取ってやればよいのです。

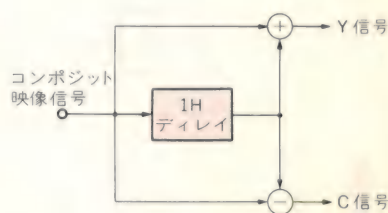
つまり、図6のような減衰特性をもつフィルタを設ければ、周波数の高い輝度成分も十分残すことができます。この特性の形が髪をすくくしに似ているところから、くし形フィルタと呼ばれています。

このような特性を実現するためには、次の二つの条件が必要です。

- (1) 色信号の位相が1Hごとに反転している。
- (2) 前後1Hの信号が類似している。

このような条件を利用して、図7に示すような操作を行うと、加算された輝度信号は、1H前後で同相で

〈図7〉  
くし形フィルタ  
原理図



すからそのまま残りますが、色信号は逆相ですので、互いに打ち消し合ってなくなります。逆に減算回路では、同相の輝度信号だけが消え去ります。その結果、YとCが分離されたことになります。

### ● CCD遅延素子によるくし形フィルタ

1Hの遅延素子には、第1章でも紹介してあるガラスを用いたものが一般的ですが、最近ではCCD(電荷

## 基礎をかため応用を考える新シリーズ

I/F ESSENCE

# A-D変換を使いこなす

パソコン信号計測ソフト&ハードの基礎と実際

〈目次〉A-D変換へのアプローチ/A-D変換の原理と技術/A-D変換用ハードウェアの実例/サポート・ソフトウェアの作成/デジタル・ストレージ・スコープへの応用/A-D装置の精度とその評価/アンチ・エリアシング・フィルタ/DMAインターフェースの実験

竹本 晃 共著 A5判, 328頁  
稲村 浩 共著 1,850円

# シリアル伝送完全マスタ

RS-232-CからRS-422/423までの詳細

〈目次〉シリアル・インターフェースとは/まず動かしてみる/RS-232-Cの構造と動作/RS-232-Cの実例/伝送コード/トラブルとその対策/RS-232-Cの応用/RS-232-Cインターフェースの製作/調歩同期式の限界と同期式/RS-232-Cの限界と発展

稲垣完治 共著 A5判, 232頁  
小野寺 徹 共著 1,440円

# IEEE-488(GPIB)とその応用

標準デジタル・バスの使い方から設計法まで

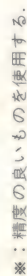
〈目次〉IEEE-488バスの基礎/安定なシステムを組み立てるには/はじめてのバスの運転/IEEE-IBの構造と動作/Basicによるプログラム/コンパイラやアセンブラによるプログラム/製作のためのアプローチ/互換性上の問題点と解決法/トラブル例とその対策

岡村 勉 著 A5判, 272頁  
1,650円

CQ出版社

〒170 東京都豊島区巣鴨1-14-2 ☎03(947)6311 振替 東京0-10665







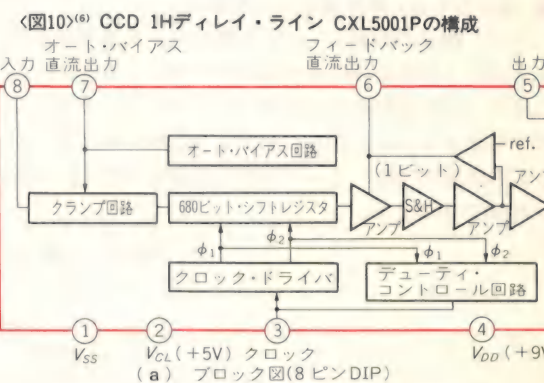
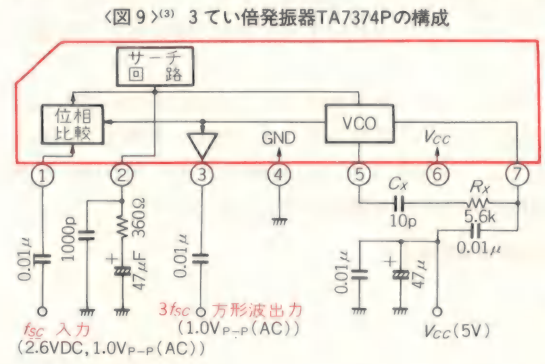
結合素子)による遅延素子が出まわっています。ここではそれを使用することにします。図8にくし形フィルタ部の回路を示します。

CCD遅延素子の特徴として、電荷の転送を行うためのクロックを供給しなければなりません。普通、クロックには10.7MHzの正弦波を使用します。

正弦波を使用するのは、転送を正確に行うのに、デューティ50%のクロックがほしいためです。この点、東芝の3てい倍発振用IC、TA7374Pを使えば、 $f_{SC}$ の3

倍の周波数である10.7MHz、デューティ50%の方形波を得ることができます。図9にこのICの特性を示します。

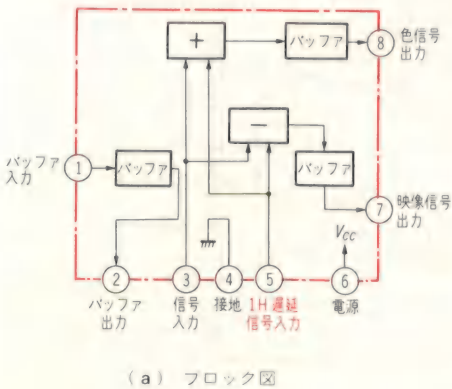
CCD遅延素子であるソニーのCXL5001Pは、外部に140nsの遅延を付加することで、合計1Hの遅延が得られます。図10にこのICの特性を示しますが、出力にはクロックの漏れ成分を除去し、入力の折り返しひずみを防ぐためのフィルタを付けることを忘れてはなりません。



項目	記号	測定条件	規格値			単位
			最小	標準	最大	
電源電流	$I_{DD}$	250kHz, 1.28V <sub>P-P</sub> 正弦波入力時	—	4	5	mA
	$I_{CL}$		—	9	11	mA
挿入利得	$IG$	250kHz, 1.28V <sub>P-P</sub> 正弦波入力 $IG=20\log$ (出力電圧 [V <sub>P-P</sub> ]/1.28 [V <sub>P-P</sub> ])	-3	0	3	dB
周波数特性	$fG$	250kHz に対する 3.5MHz での損失 $fG=20\log$ (V <sub>3.58MHz</sub> /V <sub>250kHz</sub> )	-3.0	-2.1		dB
微分利得	$DG$	5 段階段波入力 Y=1.0V <sub>P-P</sub> S 点をベクトル・スコープにより測定	—	3	5	%
微分位相	$DP$		—	3	5	deg
許容入力振幅	$V_{IN-AC}$		—	—	1.28	V <sub>P-P</sub>
ノイズ	$S/N$	S: 入力=250kHz, 1.0V <sub>P-P</sub> (出力は単位 V <sub>P-P</sub> )	55	60	—	dB
		N: 入力=交流的接地 (出力は単位 mV <sub>rms</sub> )				

(b) 主な電気的特性 ( $T_a=25^{\circ}\text{C}$ ,  $V_{DD}=9.0\text{V}$ ,  $V_{CL}=5.0\text{V}$ ,  $f_{CLK}=10.7\text{MHz}$ ,  $V_{CLK}=250\text{mV}_{P-P}$  正弦波)

＜図11＞<sup>(9)</sup> くし形フィルタ信号処理IC M51386Lの構成



項目	測定条件	規格値			単位
		最小	標準	最大	
回路電流		7.5	11	16	mA
バッファ出力電圧	SG1=200kHz 正弦波 1.5V <sub>P-P</sub>	1.3	1.5	1.7	V <sub>P-P</sub>
バッファ周波数特性	SG1=200kHz 正弦波 1.0V <sub>P-P</sub>	0.9	1.0	1.1	V <sub>P-P</sub>
	SG1=2MHz 正弦波 1.0V <sub>P-P</sub>	-2	0	+2	dB
	SG1=4MHz 正弦波 1.0V <sub>P-P</sub>	-2	0	+2	dB
加減算器出力電圧	SG1=200kHz 正弦波 1.5V <sub>P-P</sub>	1.2	1.5	1.8	V <sub>P-P</sub>
加減算器周波数特性	SG1=200kHz 正弦波 1.0V <sub>P-P</sub>	0.8	1.0	1.2	V <sub>P-P</sub>
	SG1=2MHz 正弦波 1.0V <sub>P-P</sub>	-2	0	+2	dB
	SG1=4MHz 正弦波 1.0V <sub>P-P</sub>	-2	0	+2	dB
色信号出力電圧 I	SG2=3.58MHz 正弦波 0.15V <sub>P-P</sub>	1.05	1.5	2.1	V <sub>P-P</sub>
色信号出力電圧	SG2=3.58MHz 正弦波 0.1V <sub>P-P</sub>	0.7	1.0	1.4	V <sub>P-P</sub>
ひずみ率	SG1=200kHz 正弦波 1.0V <sub>P-P</sub>	—	—	3	%

(b) 主な電気的特性 (指定のない場合は、 $T_a=25^{\circ}\text{C}$ ,  $V_{CC}=12.0\text{V}$ )



加算，減算を行うIC，三菱電機のM51386Lは，本来ガラス・ディレイを使用するためのICなので，遅延信号入力には，ガラス・ディレイでの減衰を補正する20dBの増幅器が入っています。したがって，これをCCD遅延素子に使用した場合は，レベルを合わせなければならないのですが，ピン数も少なく，値段も安いので使いやすいと思います。図11にこのICの特性を示しますが，各所にバッファが入っているのも魅力です。

### ● 完全なY/C分離を実現するために

さて，図8の回路で一応くし形フィルタはできるのですが，1H前後の信号を加え合わせているため，このままでは垂直解像度が半分になってしまいます。これを補正するには，図12に示すような操作が必要であり，各遅延時間とフィルタ特性の設計は厳密に行わなければならない。

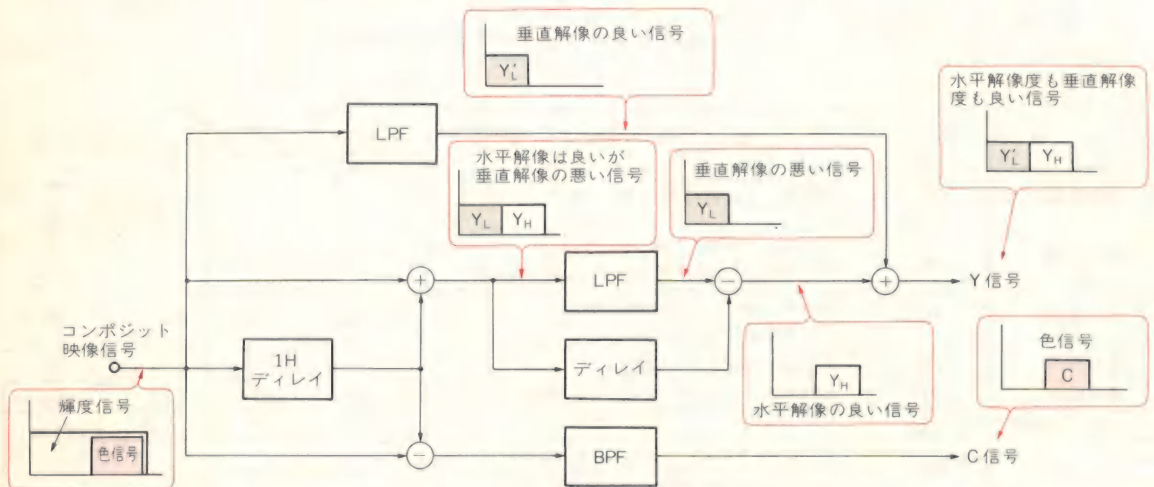
もっと簡単に補正するなら，輪郭補正機能付きの東芝のTA8620PとTL8608Pの組み合わせをお勧めします。図13にこの構成例を示します。

遅延の出る部分では，厳密に遅延時間の設定を行わないと，本来の能力を発揮してくれません。したがって，回路全体での誤差(YとCのズレ)は100ns以内に収めたいところです。そのため，ディレイ・ラインおよびフィルタは特別に注文して作る必要があります。図14にディレイ・ライン及びフィルタの仕様を示します。筆者の場合はミツミ電機㈱にて作ってもらっています。

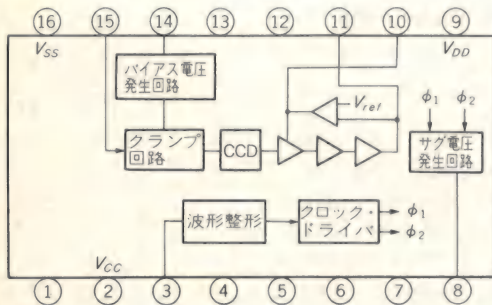
このようにして得られたYとCの信号波形を写真1に示します。

なお，フィルタを特注することができない場合には市販品の中から特性の似かよったものを捜すしかありません。参考までに，TDK㈱のカタログの中から類似性のあるものを図15に示しておきます。

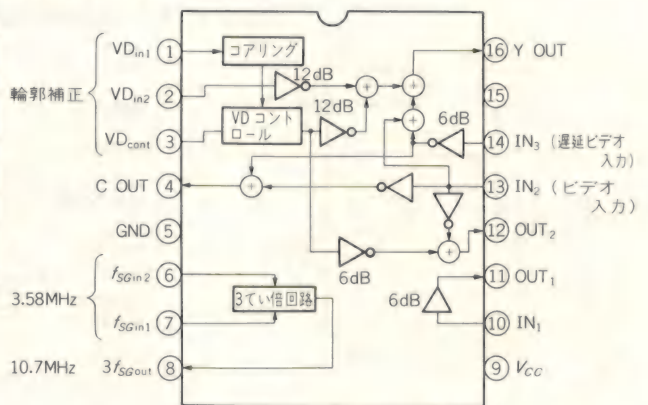
〈図12〉 くし形フィルタの垂直解像度補正



〈図13〉<sup>(3)</sup> 輪郭補正付き1Hディレイ・ライン



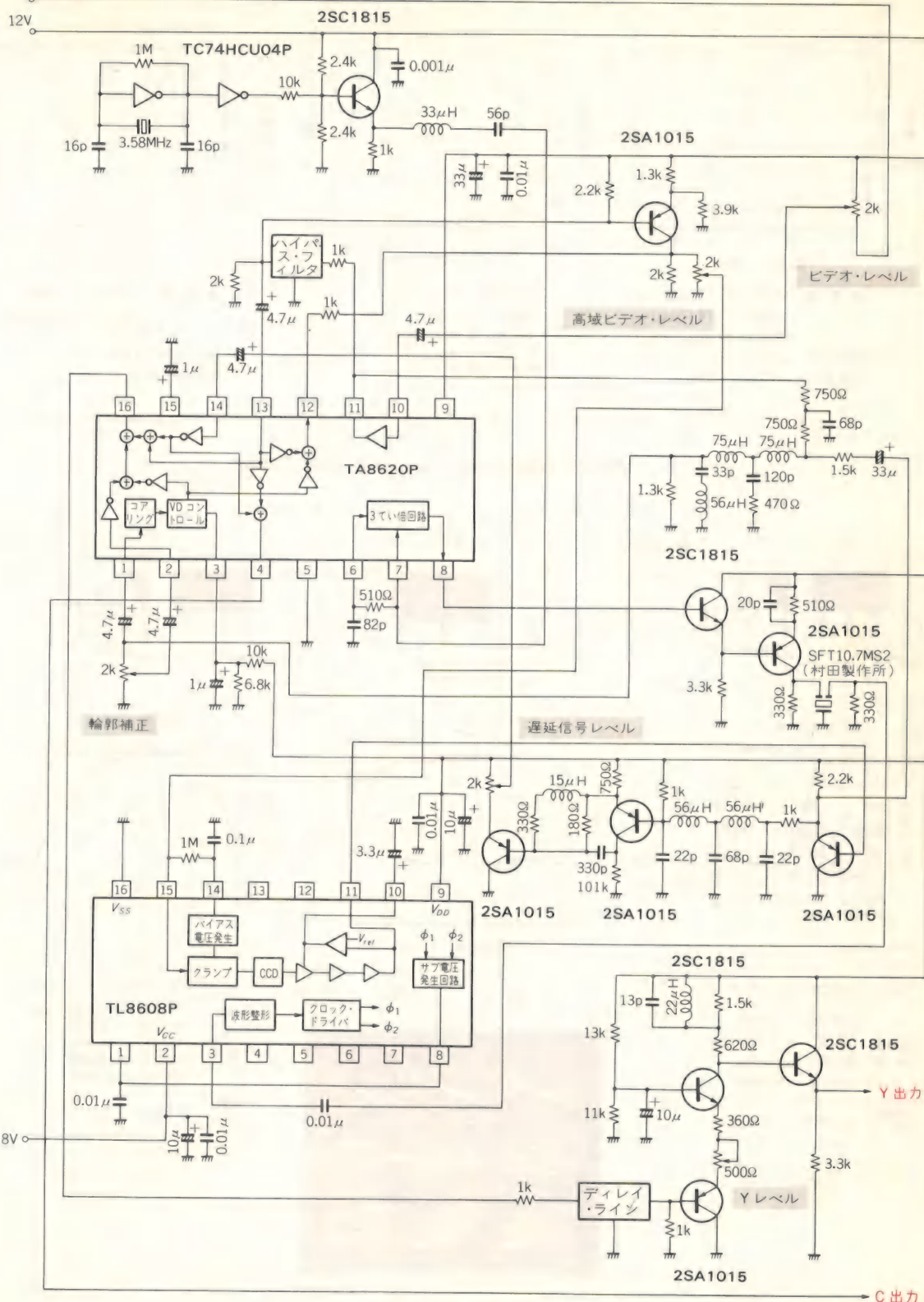
(a) TL8608P の構成



(b) TV 垂直輪郭補正用 IC TA8620P の構成



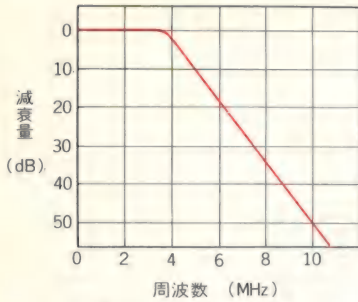
ビデオ入力



(c) TL8608PとTA8620Pの組み合わせによるCCDくし形フィルタ回路

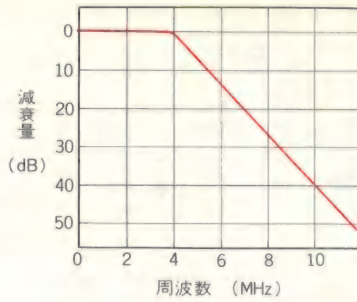


〈図14〉 Y/C分離用ディレイ・ライン、フィルタの特性例



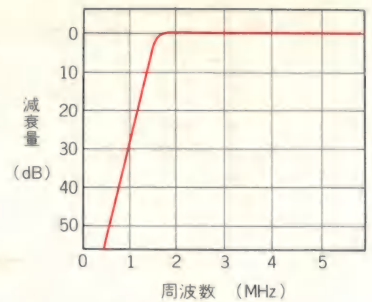
減衰量 1dB 以下 (3.58MHz)  
3dB 以下 (4.1MHz)  
50dB 以上 (10.7MHz)  
遅延時間 140ns±5%  
特性インピーダンス 1kΩ

(a) ローパス・フィルタ



遅延時間 400ns±5%  
特性インピーダンス 1kΩ  
(高域の減衰は、残留ノイズ除去のために、ノイズがなければ減衰の必要はない)

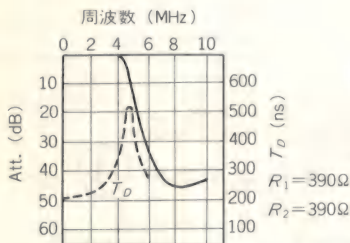
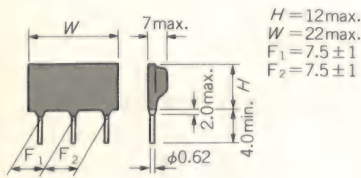
(b) ディレイ・ライン



減衰量 1dB 以下 (3MHz)  
6dB 以下 (1.5MHz)  
50dB 以上 (0.5MHz)  
遅延時間 300ns±5%  
特性インピーダンス 1kΩ

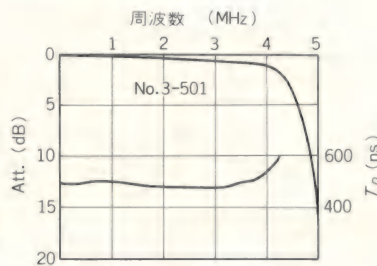
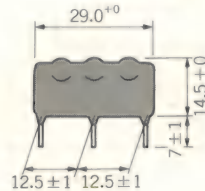
(c) ハイパス・フィルタ

〈図15〉<sup>(10)</sup> 市販のディレイ・ラインとフィルタ [TDK株]



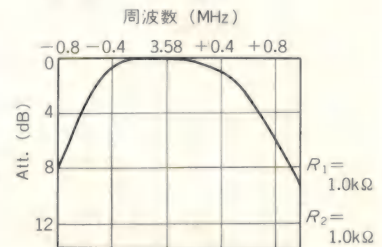
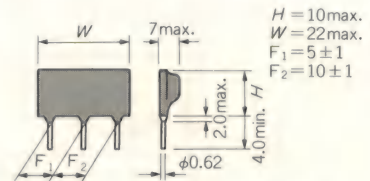
項目	規格
定格	L部: DC 30mA, C部: WV.DC.50V
周波数特性	1.0dB 以内.....3.0MHz 2±1.5dB .....4.1MHz 18±4dB .....5.0MHz 30dB 以上.....6.0MHz
群遅延特性	210±20ns .....0.2MHz

(a) ローパス・フィルタ  
SLC861-7043



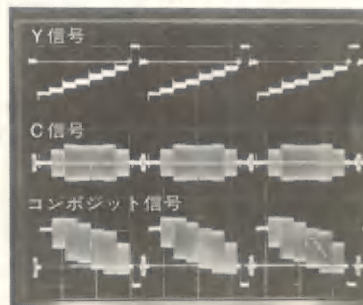
(遅延時間: 500ns)

(b) ディレイ・ライン  
DL102401D



項目	規格
定格	L部: DC.30mA, C部: WV.DC.50V
周波数特性	2.5dB 以内.....3.58MHz (Loss) 3.4±1.5dB.....2.98MHz 3.5±1.5dB.....4.18MHz 16dB 以上.....2.3MHz 15dB 以上.....5.0MHz
群遅延特性	488±30ns .....3.58MHz

(c) ハイパス・フィルタ  
SLC871-7005



〈写真1〉 Y/C分離波形  
(500mV/div, 20μs/div)



## NTSCデコーダ部の設計

この部分は、専用の良いICが出ていますので、それを使うことにします。

ソニーからは**NTSCデコーダ V7020**が出ていて、いろいろ魅力のあるICなのですが、残念ながら筆者はまだ試用していません。みなさんにはぜひ使ってみることをお勧めします。図16にその構成を示しておきます。

さて、松下電子からは、**カラー・テレビ映像、色信号処理ICのAN5310**が出ています。このICは、カラー・テレビ用なので、画像処理に使用するには精度はあまり期待できませんが、使いやすい良いICだと思います。今回はこれを使ってデコーダを組んでみます。このICの構成を図17に示します。

### ● まず同期回路を構成する

このシステムの基本的な考えとして、NTSC信号だけを入力として使いますので、**必要なパルス類はすべてNTSC信号を基に作り出さなければなりません**。そのためには、まず**同期分離を行って、水平同期信号Hと垂直同期信号Vのタイミングを拾う**ことが必要です。

図18に同期回路のブロック図を示します。AN5310

に必要なのは、**ブランキング(BL)信号**とバックボーチでクランプを行うための**ゲート・パルス(GP)**ですが、出力として**SYNC出力**も用意します。

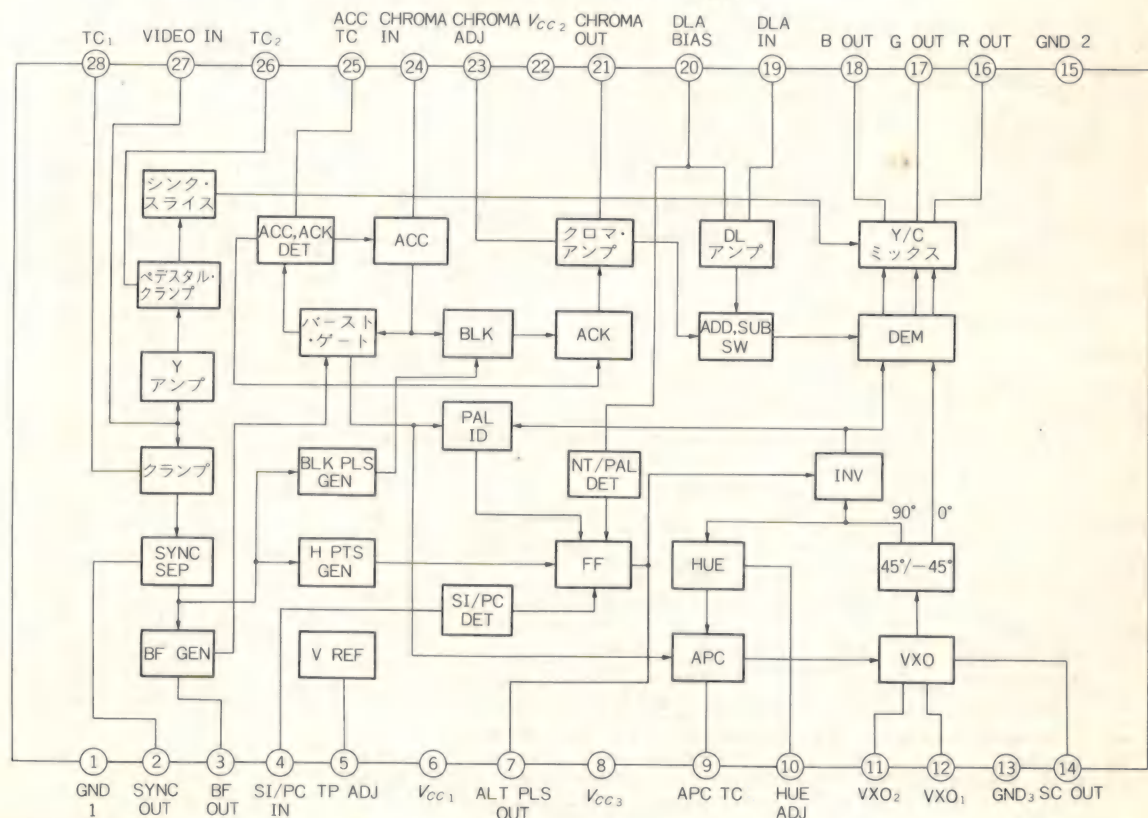
SYNCやBLを分離する場合、スレッショルド・レベルを決めて、コンパレータを使えば簡単かも知れませんが、ここで使うNTSC信号は、不特定多数のもので、必ずしもノイズがなく、しかも規格どりのものとは限りません(それを望むのはムリ)。手間はかかりますが、**一つ一つのパルスをワンショット**で作っていきます。

図19に同期回路の構成を示します。**初段で輝度信号から同期信号だけを抜きとり**、それをバッファを通してSYNC出力を得ます。そして、**次段ではH・SYNC(水平同期信号)とV・SYNC(垂直同期信号)とに分解**します。この回路で得られる波形を**写真2**に示します。

ブランキング・パルスを作るには、図20のようにそれぞれのSYNCの立ち上がりからワンショット・マルチバイブレータ(4538B)でBLの後端を得、そこから**BLの前端までの映像信号の部分に相当する長いワンショット**を作れば、それを**反転した信号がブランキング・パルス**となります。

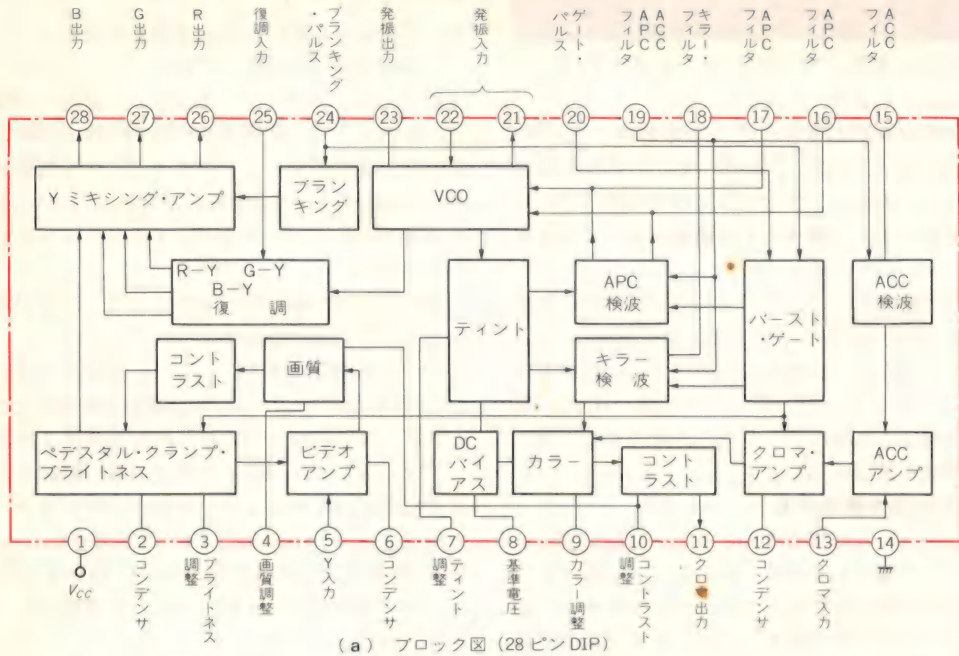
そして、**H・BLとV・BLをNANDで混合すれば複合ブランキング・パルス**となります。これらの波形を写

〈図16〉<sup>(6)</sup> NTSCデコーダ V7020の構成





〈図17〉<sup>(2)</sup> カラー・テレビ映像、色信号処理回路 AN5310の構成



項 目	条 件	min.	typ.	max.	単 位	
全回路電流	$V_{CC}=12V$	32	43	54	mA	
回路電圧	$V_{CC}=12V$	7.5 3.5	8.5 4.0	9.2 4.5	V	
出力電圧 (Burst)	レインボ 150mV <sub>P-P</sub> , カラー AUTO 中央, コントラスト max.	0.5	0.7	0.9	V <sub>P-P</sub>	
ACC 特性	レインボ 15mV <sub>P-P</sub> , バースト出力/V <sub>o(1)</sub>	0.6	0.8	1.0	times	
出力電圧 (Chroma)* <sup>1</sup>	レインボ 150mV <sub>P-P</sub> , カラー max. コントラスト max.	0.5	0.7	0.9	V <sub>P-P</sub>	
発振周波数* <sup>1</sup>	端子①⑨, 入力無信号, 標準サンプルにてトリマ設定			±150	Hz	
f <sub>osc</sub> 電源電圧依存度* <sup>2</sup>	V <sub>CC</sub> =12V±20%, 端子①⑥⑪短絡, V <sub>CC</sub> =12V に対し			±60	Hz	
f <sub>osc</sub> 周囲温度依存度* <sup>3</sup>	T <sub>a</sub> =-20~+70°C, 端子①⑥⑪短絡, IC 単体	0	-1	-2	Hz/deg.	
制御感度 (VCO)	バースト入力 0.7V <sub>P-P</sub> , 100Hz 変動に対する V <sub>16-17</sub> で換算	1.2	1.5	2.0	Hz/mV	
弁別感度 (APC)	バースト 0.7V <sub>P-P</sub> , 100Hz 変動に対する位相誤差と V <sub>16-17</sub> で換算	25	45	55	mV/deg.	
位相保持特性	バースト入力 0.7V <sub>P-P</sub> , 100Hz 変動に対する位相誤差		1.5	2.5	deg/100Hz	
APC 引き込み範囲	バースト入力 0.7V <sub>P-P</sub> , バースト周波数を変化させ測定	±450	±600		Hz	
復調出力比* <sup>4</sup>	復調入力 0.2V <sub>P-P</sub> f=3.59MHz	R 出力/e <sub>o</sub>	0.86	0.94	1.04	times
		G 出力/e <sub>o</sub>	0.25	0.30	0.35	times
復調角* <sup>4</sup>	復調入力 0.2V <sub>P-P</sub> ΔB=0deg, f=3.59MHz	R-B 位相差	94	97.5	102	deg.
		G-B 位相差	228	235	242	deg.
色差出力 (max.)* <sup>4</sup>	復調入力 1.2V <sub>P-P</sub> , f=3.59MHz, B, R 出力	4.8	5.7		V <sub>P-P</sub>	
総合色差出力* <sup>5</sup>	レインボ 150mV <sub>P-P</sub> , AUTO カラー中央, コントラスト max. R 出力	1.275	1.70	2.125	V <sub>P-P</sub>	
カラー・キラー・レベル* <sup>5</sup>	Pin⑬ 入力バースト電圧, 150mV <sub>P-P</sub> =0dB キラー動作時の減衰量	-27	-32	-40	dB	
電圧増幅度 (ビデオ)* <sup>6</sup>	スタジオ・カラー・バー白レベル 1V <sub>P-P</sub>	4.5	5.0	5.5	times	
	コントラスト max. 画質 min. *1	4.6	5.1	5.6		
周波数特性 (ビデオ)* <sup>5</sup>	正弦波 0.1V <sub>rms</sub> 入力, A <sub>V1</sub> が -6dB となる入力周波数, 画質 min. 出力 B	5	6		MHz	
DC 伝送量	ビデオ入力 1V <sub>P-P</sub> (ステア・ステップ) APL10~30% B 出力	68	75	82	%	
原色出力電圧 (max.)	V <sub>3</sub> =12V での R, G, B 各出力電圧	7.0			V	
微分利得* <sup>6</sup>	ビデオ入力 1V <sub>P-P</sub> (ステア・ステップ 3.58MHz) APL50%, コントラスト max. 画質 min			5.0	%	
復調出力直流電圧* <sup>7</sup>	ブライト V <sub>3</sub> =9V, コントラスト max. 無信号時, VCO 発振, RGB 各出力	2.8	3.5	4.2	V	
復調出力各端子間直流差電圧	V <sub>26</sub> =3.5V, VCO 発振, R, G, B 各出力		0	300	mV	
ΔE <sub>X-Y</sub> 電源電圧依存度* <sup>8</sup>	V <sub>CC</sub> =12V±20%, V <sub>26</sub> =3.5V (V <sub>CC</sub> =12V), V <sub>CC</sub> =12V に対して		0	±60	mV	
ΔE <sub>X-Y</sub> 周囲温度依存度* <sup>8</sup>	V <sub>26</sub> =3.5V (T <sub>a</sub> =25°C), T <sub>a</sub> =-20~70°C, T <sub>a</sub> =25°C に対して		0	±60	mV	

\*1  $f: 357945Hz$  からの偏差    \*2 変動の最大幅    \*3 変動の最大幅  $90^\circ C$

\*4  $V_{26}=3.5V$ , ブライト VR 設定, プランキングなし    \*5  $V_{26}=3.5V$  ブライト VR 設定

\*6 R, G, B 各出力, 端子②⑥ベDESTAL 2V, ブライト VR 設定    \*7 プランキングあり    \*8 プランキングなし

(b) 主な電気的特性



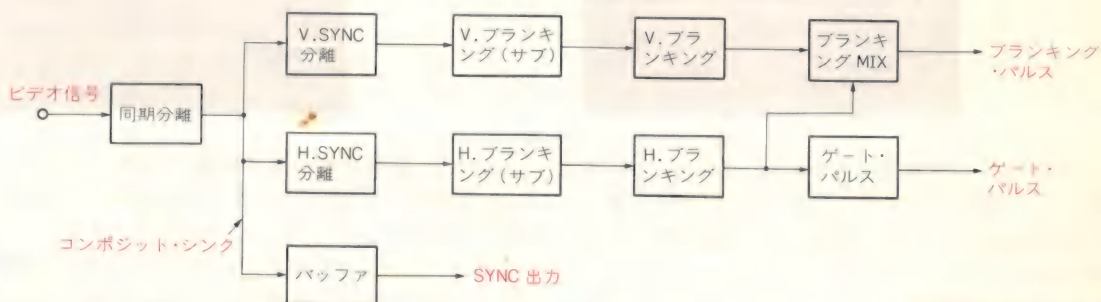
真3に示します。

ゲート・パルスはクランプに使うので、バック・ポーチ内なら良いのですが、後で利用できるかも知れませんが、位置・幅をバーストに合わせることにしま

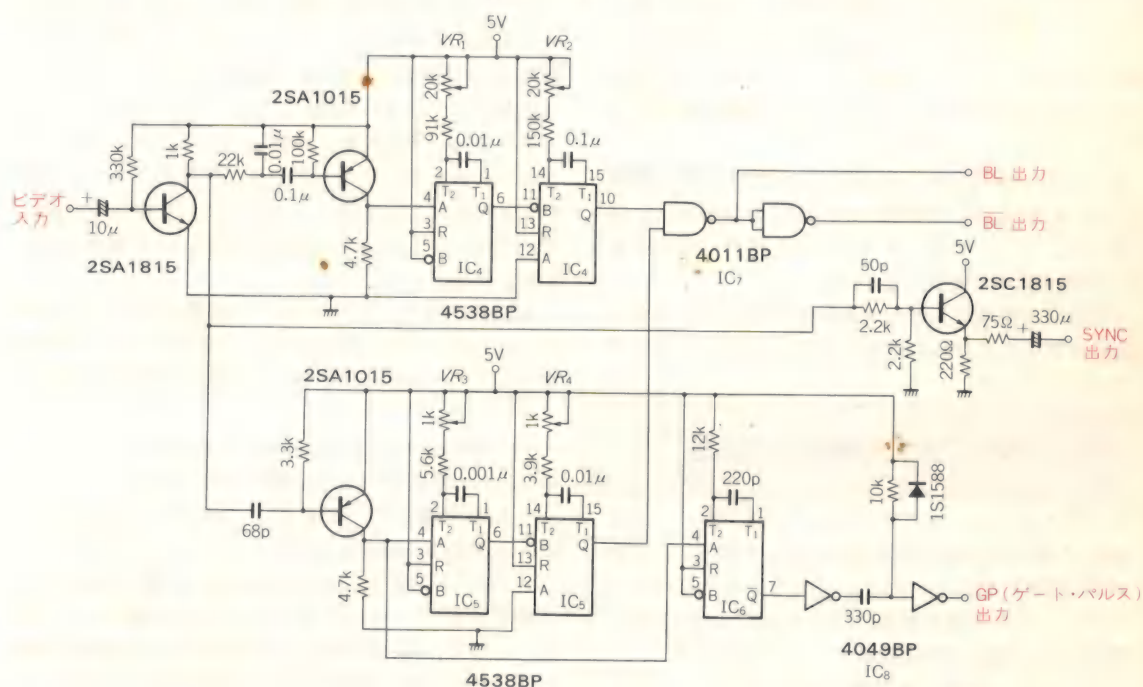
す。このような信号をバースト・フラッグ(BF)といいます。

ゲート・パルスもワンショットで作ります。これを写真4に示します。

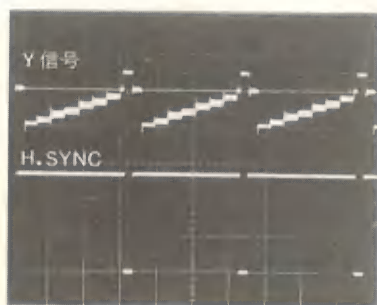
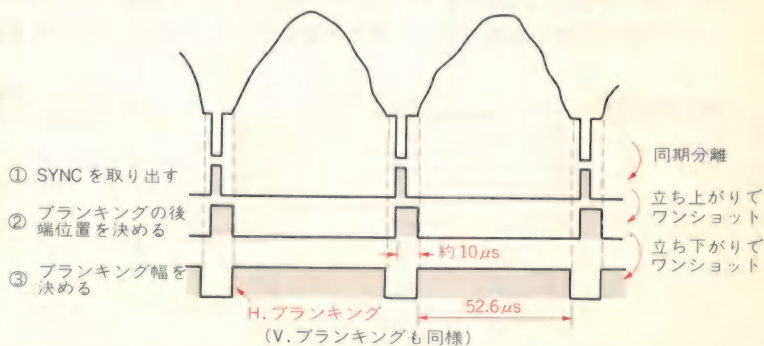
〈図18〉 同期回路ブロック図



〈図19〉 同期回路



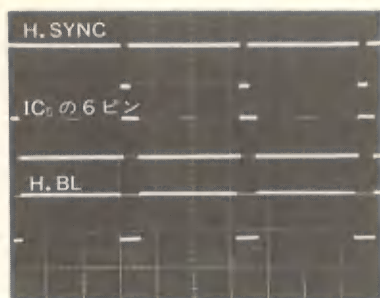
〈図20〉 ブランキング・パルスの作り方



〈写真2〉 同期分離によるH.SYNCの波形  
(上: 500mV/div, 下: 2V/div,  
20μs/div)



〈写真3〉  
3Hブランキング  
波形(5V/div,  
20 $\mu$ s/div)



このように、ワンショットを連続させると、元の信号が少しでも乱れるとすべてのタイミングが狂ってしまう危険性がありますが、**任意にパルス幅が設定できる**のが魅力です。

### ● NTSC→RGB変換

図21にAN5310によるNTSC→RGB変換の回路図を示します。AN5310には、図19で作った**ブランキング・パルス**と**ゲート・パルス**、図8で作った**Y信号**と**C信号**、各種コントロールを行うためのコントロール電圧を入力し、外付けクリスタルで発振回路を構成します。

コントロールはすべてDC電圧ですので非常に簡単ですが、むやみに可変範囲を広げることは望ましくありません。また、**オート・スイッチを付けることにより、調整を簡単にすることができます。**

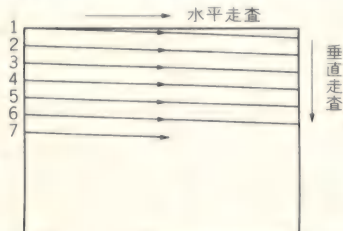
Y信号は負極性で入力しますので注意してください。C信号はひずみのないものを入力しないと、色のバランスがくずれます。

## NTSC方式の同期と走査法

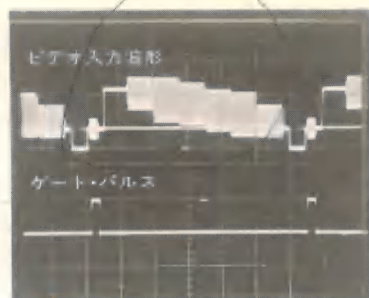
テレビジョン・システムでは、カメラのレンズを通して得た光の像を電気信号に変えて伝送し、それを再びCRT上に光の像として再現しなければなりません。この伝送を簡単に行うためには、一系統の電線または電波で間に合うことが必要です。

そのため、画像を送る側では光の像をたくさんの明暗の点(画素という)に分解し、**順序よく送り出します。**受ける側では、それを**正しい順序で組み立てて一つの像を再現**します。この一連の作業を走査と

〈図A〉 順次走査  
(ノン・インタレース)



〈写真4〉  
ゲート・パルス波形  
(上: 500mV/div,  
下: 5V/div,  
10 $\mu$ s/div)



このICは副搬送波の発振回路を内蔵していて、**3.579545MHzのクリスタル**と、**安定用のコンデンサ**だけですみます。したがって、普通はこのままでよいのですが、図のようにバッファでインピーダンス変換を行って出力すれば、ほかのシステムにも使えて回路の節減に役立ちます。

また、AN5310の端子24にブランキング・パルスをそのまま入力すれば、きれいなブランキングの付いたRGB出力が得られますが、本来のペダスタル・レベルは無視されてしまいます。

それを防止するため、本機ではブランキング・パルスのレベルを落として入力し、不完全なブランキングのまま出力しています。したがって、この回路の後にブランキング・ミックス回路が必要になる場合もあります。

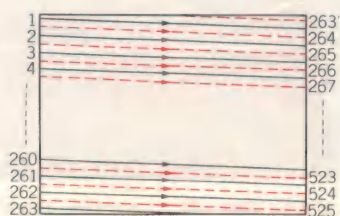
またIC内部で、ブランキング期間中にクランプをかけていますので、そのタイミングにバースト信号があると**ミス・クランプ**を起こし、R・G・BのDCレベル

いい、NTSC方式として規格化されています。

一枚の画像をテレビ管面上に映すには図Aのように、水平走査と垂直走査を行います。

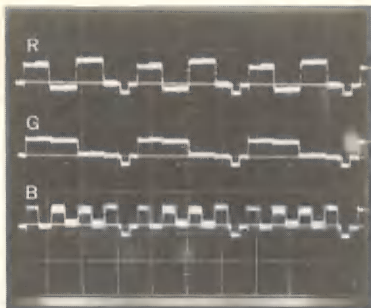
しかし、NTSC方式はこれと異なり、図Bのような**飛び越し走査(インタレース)**という方式を採用しています。これは、初めの画面では奇数番目の走査線を走査し、次の画面では偶数番目の走査線を走査します。このように**2回の走査で一つの画像を構成**することで、画面のちらつき(フリッカ)を少なくし、同時に前の走査でできた走査線の隙間を次の走査線が埋めるので、

〈図B〉 飛び越し走査  
(2:1インタレース)

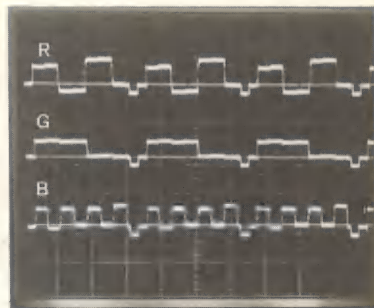




〈写真5〉  
RGB出力波形  
(5 V/div,  
20μs/div)



〈写真6〉  
トラップでノイズ  
除去後のRGB波形  
(5 V/div,  
20μs/div)



に差ができてしまいます。そこで、**クロマ信号からバースト信号を消去する回路を端子25に接続してあります。**

得られるRGB信号は、**写真5**のようになります。このままでは**副搬送波がかなり残留**していますので、**実際にはトラップなどで除去したほうが良い**でしょう。ただし、安易に設計されたトラップではせっかくし形フィルタに残された高域成分も一緒に除去されてしまいます。慎重に選んでください。ここでは村田製作所のTPS3.58MJという標準のトラップを使用しています。

以上の出力波形を**写真6**に示します。

#### ● 調整の方法

調整に必要な機材。

- (1) オシロスコープ
- (2) モニタ・テレビ(RGB入力付き)
- (3) カラー・カメラ
- (4) 電源(DC12V, 500mA程度)

(5) カラーバー・チャート。なければ色彩豊かなポスト、色紙など

オシロスコープは50MHz以上のもの、モニタ・テレビは家庭用のAVテレビでも良いのですが、RGB入力があるBNCコネクタのものの方が使いやすいと思います。筆者はソニーのPVM1371Qを使っています。

また、カラー・カメラは、このような実験には色の再現性が良いことが要求されますので、ここではトキナー光学のCCDカラー・カメラ、CP8001を選びました。もちろん手持ちのVTR用カメラでもかまいません。表1にCP8001の仕様を示しておきます。

調整は、カメラのホワイト・バランスを合わせてから、出力波形を見て、

- (1) Vブランキングの後端を $VR_1$ で合わせる。
- (2) 同前端を $VR_2$ で合わせる。
- (3) Hブランキングの後端を $VR_3$ で合わせる。
- (4) 同前端を $VR_4$ で合わせる。(以上図19参照)
- (5) ブライツネス、画質、ティント、カラー、コント

見かけ上の解像度を向上させる効果があります。

当然、**図A**のような順次走査方式の2倍の走査線数が得られ、**525本の走査線で一つの画面を構成**します。

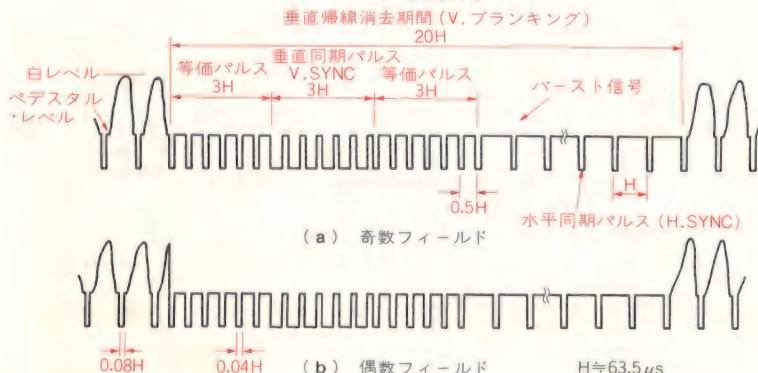
このような走査方式を常に安定して継続させるのが同期信号の役目です。**図C**にNTSCの垂直・水平同期信

号を示します。

垂直同期信号に挿入されている等価パルスは、飛び越し走査を正確に行わせるためのものです。

NTSC方式の諸元を表Aに示しておきます。

〈図C〉 NTSCの同期信号

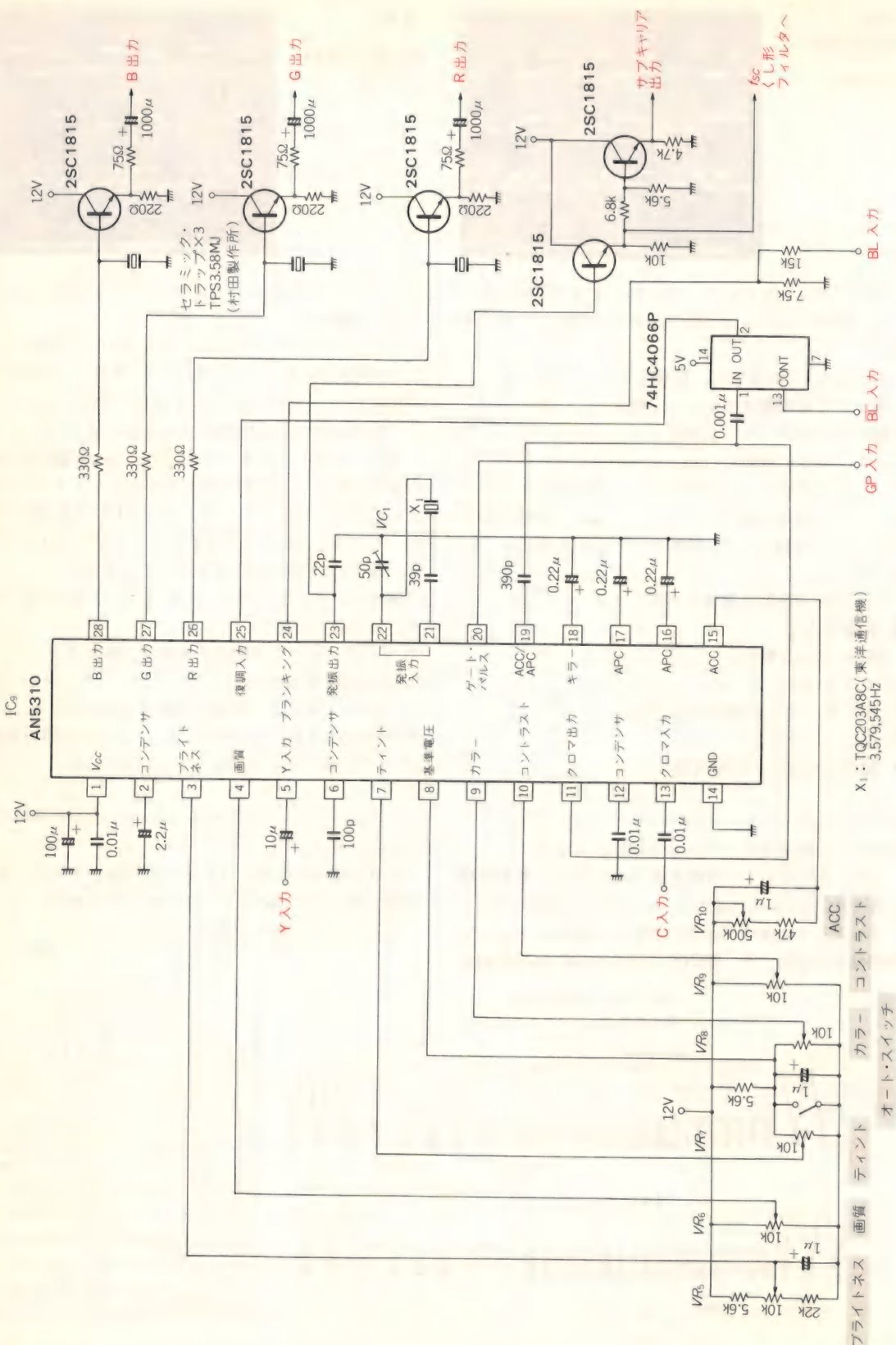


〈表A〉 NTSCの諸元

項目	規格
走査線数	525本
水平周波数	15,734.264Hz
水平周期 [H]	63.492μs
水平ブランキング幅	10.9±0.2μs
水平SYNC幅	4.7±0.1μs
垂直周波数	59.94Hz
垂直周期	16.667ms
垂直ブランキング幅	20H
垂直SYNC幅	3H
毎秒像数	29.97枚
色副搬送波周波数	3,579.545Hz



図21> デコーダ回路



X1: TQC203A8C(東洋通信機)  
3.579,545Hz



(2)\*松下電子, National Electronic Components 6 半導体集積回路〈リニアIC編〉.

(3)\*東芝，集積回路技術資料，TA7374P，TL8608P，TA8620P。

(4)\*日本真空光学, カタログ.

(5)\*モトローラ, MC1377カタ

(6)\*ソニー, Semiconductor IC Data Book 1986, カメラ・ビデオレコーダ編.

(7) テレビジョン学会：テレビジョン・画像工学ハンドブック，  
オーム社，第一版，1980年12月。

(8) 日本放送協会；カラーテレビジョン，日本放送出版協会，昭和43年。

(9)\*三菱電機, 三菱半導体テレビ/ビデオ編データブック.

(10)\*TDK, インダクタ・フィルタ・カタログ

## ◆参考・引用文献◆

- (1) 赤羽根 仁：映像信号のインターフェース技術，トランジスタ技術，1986年6月号。

▶▶▶▶▶▶▶▶▶▶好評のCORE BOOKSシリーズの第4弾!!



マイコン画像処理ハード&amp;ソフトの設計・製作▶▶畔津明仁 著

基礎からの映像信号処理

映像信号処理 第2巻 1.1 画素の表現と色

朝倉明仁



第1章 入門・映像信号処理、第2章 実験用・画像処理装置の設計と製作、第3章 映像信号処理ハードウェアの研究、第4章 基本画像処理ソフトウェアの実例(2値化/強調/重心測定/エッジ抽出/パターン・マッチング/階調補正/フーリエ変換ほか)

CQ出版社



## NTSCエンコードの設計・製作



角田和宏  
渡辺成治

RGB信号を一般のテレビでモニタするには、RGB→コンポジット信号コンバータ、あるいはさらに電波に乗せるためのRFモジュレータが必要です。ここではこの両方の技術について紹介します。

第2章でも紹介したように、NTSCコンポジット信号は、一つの線に複合した情報をつめこんでいますので、画像信号の伝送という点では大変有効なものです。しかし、画質という点ではRGB方式にくらべると劣ってしまいます。したがって、最近のニューメディア対応機器では、RGB21ピン・マルチコネクタなどを使用する傾向が多くなっています。

しかし、そうはいつても身近にあるビデオ機器の多くは依然としてコンポジット信号を使用する傾向が多く、RGB方式の信号を身近なビデオ機器に接続する場合には、RGB信号をNTSCコンポジット信号に変換しなければなりません。このような機器をNTSCエンコーダと呼んでいます。この具体的な設計法を紹介することにします。

なお、NTSCコンポジット信号の授受は、ベース・バンドのコンポジット信号そのものを扱うほかに、テレビの空きチャンネルに電波として挿入する方法もあります。このRFモジュレータの設計についても併わせて紹介することにします。

### NTSC信号ができるまで

NTSCエンコーダ自体はIC 1個で簡単に構成する

ことができますが、ここでは少し原理に立ちかえて調べてみることにします。

#### ● ビデオ用カメラでのNTSC信号生成

ここでは、ビデオ用カメラでのNTSC信号ができるまでの過程を図1を用いて説明します。

カメラのレンズを通った光は、三色分解プリズムによってR・G・Bの三原色に分けられ、それぞれの撮像管に結像します。

このプリズムには、図2のように反射面にダイクロイック・コートと称する多層膜が蒸着してあり、赤または青のように、ある特定の波長のみを反射させることができ、これによって三色分解ができます。図(b)にその分光特性例を示します。

撮像管で光電変換されたビデオ信号は、プリアンプ、クランプ、γ補正などの各種信号処理が行われ、それぞれR・G・B信号になります。そしてマトリクス回路によって、RGB原色信号から輝度信号(Y)と色差信号(R-Y, B-Y)を作ります。

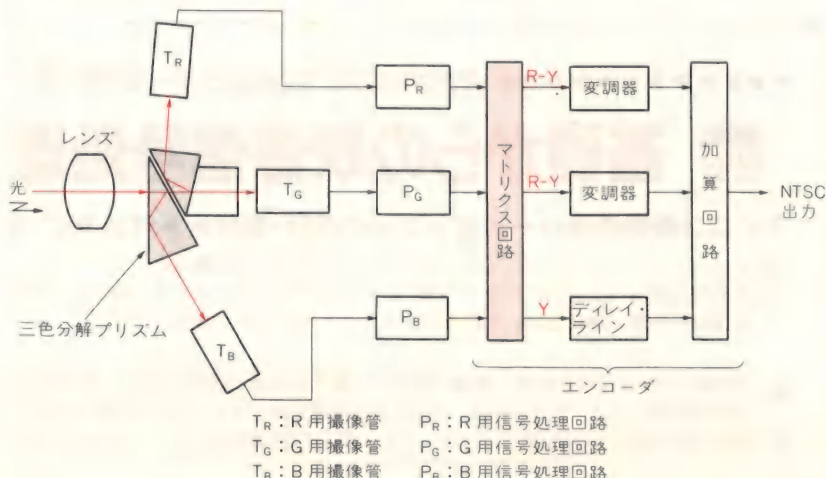
#### ● 輝度信号と色差信号との関係

輝度信号はR, G, Bの各成分が一定の比率で合成された信号であり、その関係は(1)式で表されます。

$$E_Y = 0.30E_R + 0.59E_G + 0.11E_B \dots\dots\dots(1)$$

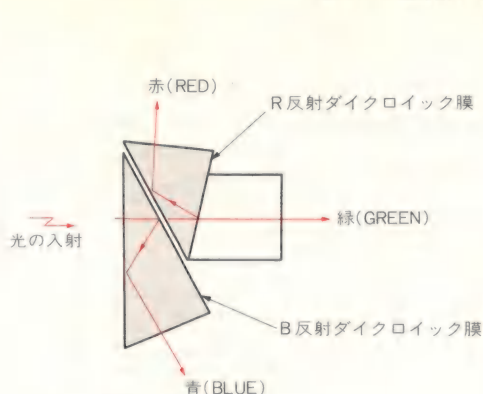
$E_Y$  : 輝度信号成分

〈図1〉 NTSC信号のできるまで

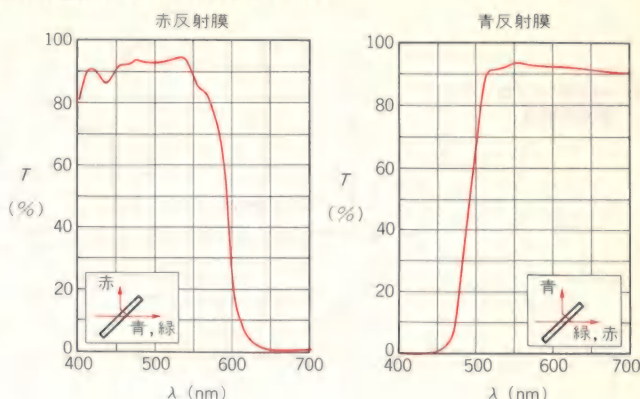




〈図2〉 三色分解プリズム(3種類のプリズムからできている)

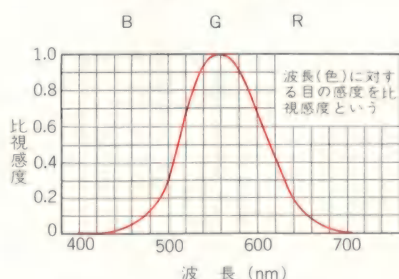


(a) 構成図



(b) 分光特性

〈図3〉 視感度曲線



$E_R$ : 赤信号成分  
 $E_G$ : 緑信号成分  
 $E_B$ : 青信号成分

このような比率は、人間の視感度を基に決められています。図3のように人間の目が明るさを感じる時、緑(555nm)を最も明るく感じ、次に赤、青の順になることと合致しています。

色差信号は(1)式より、

$$\begin{aligned} E_R - E_Y &= E_R - (0.30E_R + 0.59E_G + 0.11E_B) \\ &= E_R(1 - 0.30) - 0.59E_G - 0.11E_B \\ &= 0.70E_R - 0.59E_G - 0.11E_B \quad \dots\dots(2) \end{aligned}$$

同様に、

$$E_B - E_Y = -0.30E_R - 0.59E_G + 0.89E_B \quad \dots\dots(3)$$

となります。

ちなみにこの色差信号は、被写体が無彩色(白色のようにRGB各成分が均等に含まれている)の時には0になるようになっています。例えば(2)式で、 $R=G=B=1$ とすると、

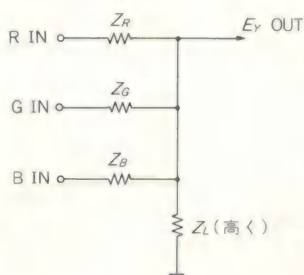
$$E_R - E_Y = 0.70 - 0.59 - 0.11 = 0$$

となり、各原色信号が輝度信号と等しいことを意味します。

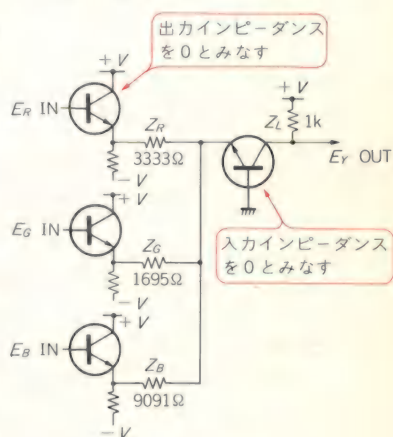
#### ● マトリクス回路の構成方法

RGB信号から、 $E_Y$ 、 $E_R - E_Y$ 、 $E_B - E_Y$ などの信号

〈図4〉 抵抗マトリクス



〈図5〉 実用的なマトリクス回路



を得るためのマトリクス回路は、原理的に図4のような抵抗の組み合わせで構成できます。各抵抗の定数 $Z_R$ 、 $Z_G$ 、 $Z_B$ は前記の比率になっています。

実用的なマトリクス回路としては、エミッタ・フォロウ回路とベース接地アンプを組み合わせた図5の回路を示します。

エミッタ・フォロウの出力インピーダンス及びベース接地アンプの入力インピーダンスを0と考え、

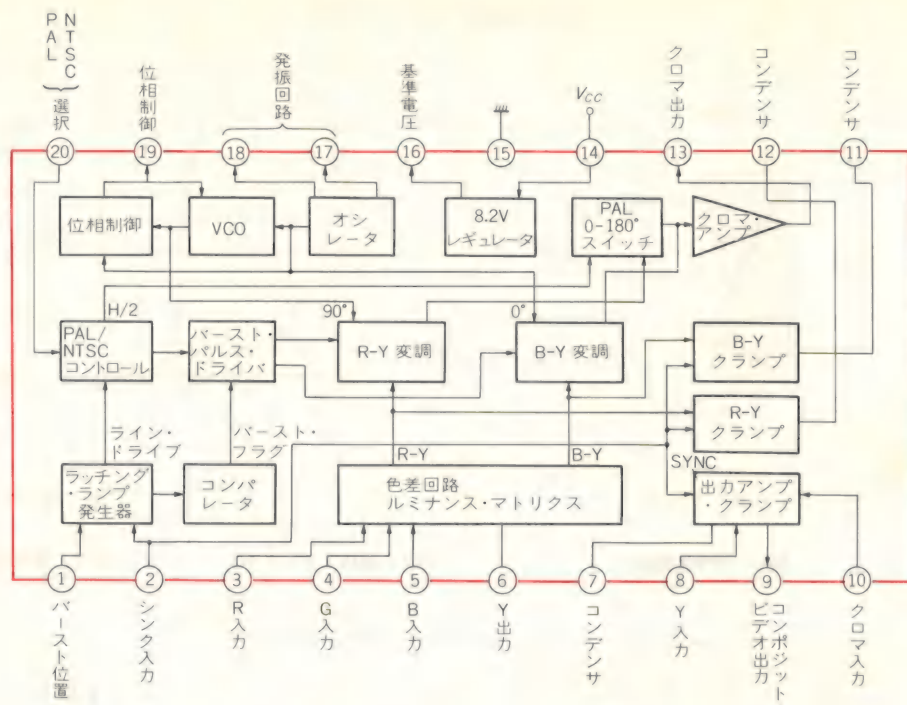
$$\begin{aligned} E_Y &= \frac{Z_L}{Z_R} + \frac{Z_L}{Z_G} + \frac{Z_L}{Z_B} \\ &= 0.30E_R + 0.59E_G + 0.11E_B \end{aligned}$$

となりますので、アンプ機能をもったマトリクス回路が構成できます。 $E_R - E_Y$ や $E_B - E_Y$ も同様に考えることができます。

このようにして得られたR-Y信号とB-Y信号とで、 $f_{sc} = 3.579545\text{MHz}$ の色別搬送波(サブキャリア)を変調したものが色信号になります。ちなみに、R-YとB-Yとの間には、 $90^\circ$ の位相差をもたせています。



〈図6〉 MC1377Pブロックと電氣的特性



推奨動作条件

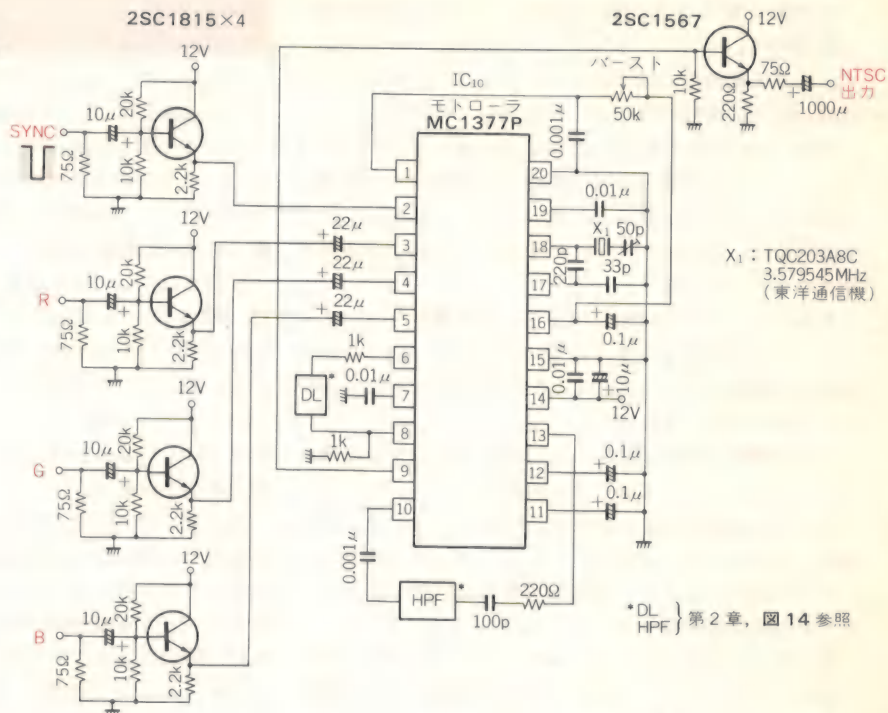
供給電圧	12 ± 1.2	V <sub>dc</sub>
シンク、先頭レベル	-0.5 ~ +1.0	V <sub>dc</sub>
シンク、プランキング部レベル	+1.7 ~ +8.2	V <sub>dc</sub>
R, G, B 飽和入力	1.0	V <sub>p-p</sub>

電氣的特性 (V<sub>CC</sub> = 12V<sub>dc</sub>, T<sub>a</sub> = 25°C)

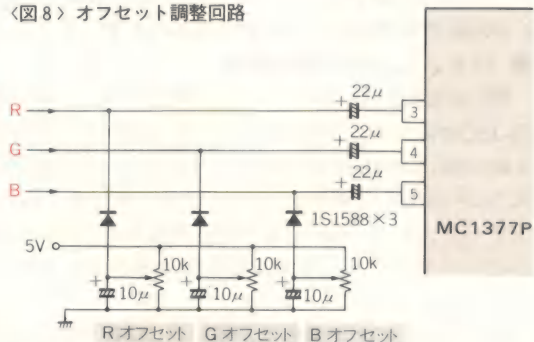
項 目	ピン No.	min	typ	max	単 位
消費電流	14	20	32	40	mA <sub>dc</sub>
発振回路出力振幅	18	—	0.5	—	V <sub>(P-P)</sub>
外部サブキャリア入力	17	—	0.25	—	V <sub>rms</sub>
サブキャリア：入力抵抗	17	—	5.0	—	kΩ
入力容量		—	2.0	—	pF
変調位相 (R-Y) ~ (B-Y)	—	85	90	95	Degrees
(R-Y) 位相調整範囲	19	—	0.25	—	Deg/μA
100% 変調時 RGB 入力	3, 4, 5	—	1.0	—	V <sub>(P-P)</sub>
RGB 入力抵抗	3, 4, 5	—	10	—	kΩ
入力容量		—	2.0	—	pF
シンク・スレッショルド・レベル	2	—	1.7	—	V
シンク入力抵抗	2	—	10	—	kΩ
クロマ出力レベル	13	—	1.0	—	V <sub>(P-P)</sub>
クロマ出力抵抗	13	—	50	—	Ω
クロマ入力レベル	10	—	0.7	—	V <sub>(P-P)</sub>
クロマ入力抵抗	10	—	10	—	kΩ
入力容量		—	2.0	—	pF
コンポジット出力	9	—	0.6	—	V <sub>(P-P)</sub>
		—	1.4	—	
		—	1.7	—	
		—	0.6	—	
出力インピーダンス	9	—	50	—	Ω
ビデオ帯域幅	9	—	8.0	—	MHz
キャリア・リーク	9	—	20	—	mV <sub>(P-P)</sub>



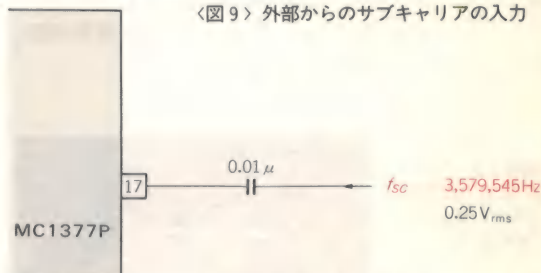
〈図7〉 NTSCエンコーダの構成



〈図8〉 オフセット調整回路



〈図9〉 外部からのサブキャリアの入力



## NTSCエンコーダの設計

市販のNTSCエンコーダ用ICは、RGB信号をマトリクス回路でNTSC信号に組み立てる部分、色差信号を作る部分、オーディオ(音声)信号を付加する部分などが1チップで実現できるようになっており、最近の良いICが出まわっています。ソニーのV7040、ロームのBA7230LSなどがありますが、今回はモトローラのMC1377Pを使ってみました。図6にMC1377Pの構成を示します。

### ● MC1377Pを使う

このICは、エンコードに必要な処理をほとんど内部で行ってくれますので、非常に簡単に回路を構成できます。内容的には、デコードの逆の作業を行うと考えてよいでしょう。

図7にNTSCエンコーダの実用回路を示します。RGB入力端子は、サグが出ないように比較的大きい容量のコンデンサでカップリングしています。規定の出力を得るためのRGB入力1.0V<sub>P-P</sub>である点は念頭においてください。

また、ここでそれぞれの信号がDCオフセットをもっていると、出力でペダスタル・レベルが上がってしまうばかりでなく、**正確な色再現ができません**。したがって、図8のように、**オフセット調整回路**を設けたほうが良い場合もあります。

発振回路は内蔵していますので、クリスタルをつなげばこれ単独の副搬送波は得られます。もし、ほかの回路、例えば第2章で述べたNTSCデコード回路と組み合わせるようでしたら、第2章の図21にも示してあるトランジスタ・バッファを通して、ほかの副搬送波を図9のように入力することもできます。部品代の節約になると同時に、位相の不一致によるひずみやノイズを防止することができます。



## ● ICを使う時の注意事項

このICを使ううえで最も重要なことは、輝度信号も色信号も、いったんICの外に出してレベルを合わせ、もう一度内部に取り込みますので、その部分での処理の仕方でも性能の良しあしが決まることです。

また、色信号には絶対に直流成分が乗ってはいけません。ここで直流成分が乗っていると輝度信号に影響を及ぼし、明るさが狂ってしまいます。したがって、本回路ではハイパス・フィルタを使って不要な直流成分をカットしています。

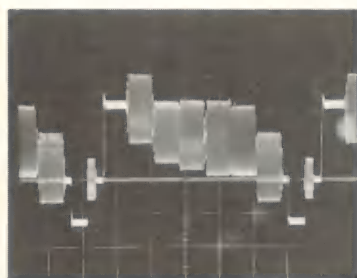
さらに、ここでの振幅レベルで、色の濃さとバースト・レベルが決定されますので、バースト・レベルがNTSCの規格に合うよう、またひずみが出ないように設計しなければなりません。

一方の輝度信号ですが、ここではレベルを半分に落とすことになっています。また色信号のローパス・フィルタでの遅延時間に合わせて、ディレイ・ラインを構成しなければなりません。したがって、くし形フィルタでの説明のように、周波数特性の良いディレイ・ラインが必要になります。

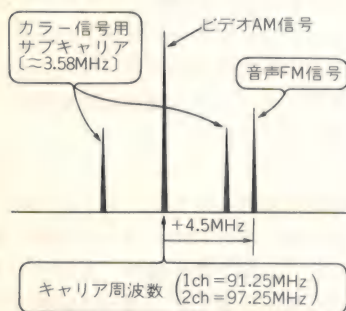
規定の入力を与えたとき、NTSC出力は $2.0V_{P-P}$ ですので、出力バッファを通して $75\Omega$ で終端すれば、写真1のように $1.0V_{P-P}$ のNTSC信号がでかかります。

〈角田和宏〉

〈写真1〉  
NTSC出力波形  
(500mV/div,  
10 $\mu$ s/div)



〈図10〉ビデオと音声信号のRF割り当て



## RFモジュレータの設計

マイコンやパソコン、あるいはビデオ機器の信号をテレビ(CRT)に表示する簡単な方法として、RFモジュレータを使う方法がありますが、ここではNTSCコンポジット信号と音声信号とを放送波に変換するRFモジュレータの回路について紹介します。

### ● テレビ放送波の規格

変調されたビデオと音声信号、つまりテレビ放送波は、図10のように公的に定められた周波数構成となっています。日本の場合、キャリアの周波数は東日本が2ch、西日本では1chが空きチャンネルとなっており、このチャンネルがRFモジュレータ用として使用できるわけです(2chか1chかはRFモジュレータ側で切り替える必要がある)。

例えば、1chの場合は91.25MHzがキャリアで、音声信号は4.5MHzでFM(周波数変調)されたあと、キャリアと周波数ミキシングされます。そして、さらにキャリアとAM(振幅変調)されたビデオ信号とが合成され、図10のような周波数分布に設定されます。

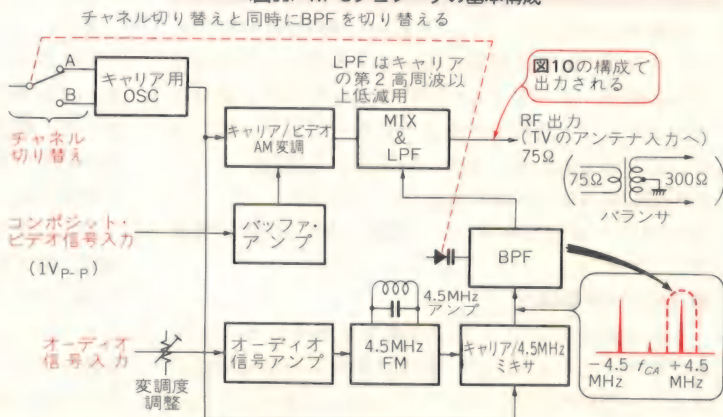
なお、テレビ受像機のビデオ信号は、IFアンプの次段でAM検波すると再生されますが、音声信号は4.5MHzの音声IF信号として出力されます。

### ● RFモジュレータの基本構成

図11がRFモジュレータの基本構成です。キャリア用OSCは切り替え可能になっており1ch=91.25MHz、2ch=97.25MHzに設定されています。ビデオ信号の入力レベルは、一般的に $1V_{P-P}$ ( $75\Omega$ 、または1k $\Omega$ インピーダンスで、両者共に水平同期信号から白レベルのピークまでのコンポジット信号)を条件としています。

日本では、ビデオ信号は振幅変調(AM)され、音声信号は、第一段階に4.5MHzで周波数変調(FM)され、そして第二段でキャリア周波数とミキシングさ

〈図11〉RFモジュレータの基本構成





れ、図11のようにキャリア周波数±4.5MHzの信号となります。なお次段のバンドパス・フィルタ(BPF)でキャリア周波数+4.5MHzの信号を再使用します。ただし、BPFはchごとにその周波数特性を切り替える必要があり、通常はLC共振回路のCをチャンネル切り替えと連動スイッチで切り替えます。

変調されたビデオ信号と音声信号は、お互いにミックスされますが、キャリア信号の第2高調波を低減するためにローパス・フィルタ(LPF)を通したあと、やっとテレビ・アンテナへ入力すべきRF信号となります。

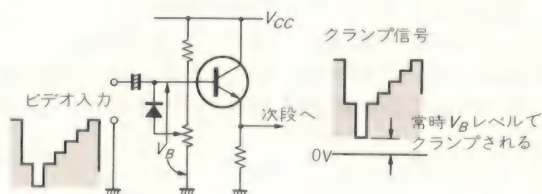
テレビのアンテナ入力は、75Ω系アンバランス型と300Ω系バランス型がありますが、300Ω系の場合は、この出力にバラサを付加します。

以上が基本的なRFモジュレータの構成ですが、ローコストのパソコンやゲーム専用コンピュータなどは、これらをトランジスタ1〜2個で構成しており、変調回路のリニアリティなどに限界があるようです。

### ● ビデオ信号の変調に要求される主な特性

ビデオ信号(コンポジット信号)は、黒レベルから白レベルの間はきわめて急峻な信号の動きとなります。

〈図12〉 水平同期信号レベルでのクランプ



したがって、同期信号そのものも信号と一緒にレベルがふらつくことが予想されます。そこで、図12の回路例のように、ビデオ信号が水平信号レベルでクランプすることがありますが、こうすると変調回路も安定なレベルで動作できるようになります。

ところで、ビデオ信号はAM(振幅変調)に際して次の3点に注意する必要があります。

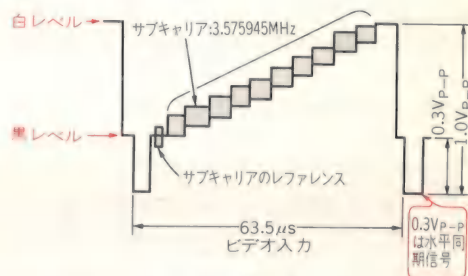
- (1) 水平同期信号の再現性
- (2) 微分利得
- (3) 微分位相

これらは計測用の標準受信機で測定します。図13が実際の測定時のビデオ信号の状態です。

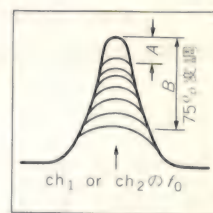
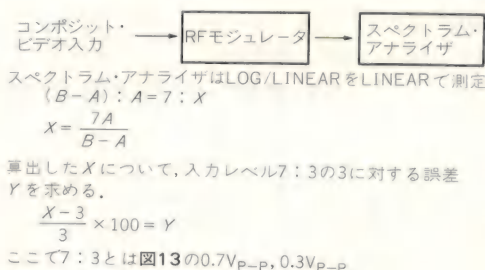
(1)の水平同期信号の再現性については、テレビ受像機での同期分離回路への必要条件となります。通常は、図14の測定において数%までは許されています。

(2)の微分利得は、白レベルから黒レベルまでの間のカラー信号用サブキャリアの信号レベルを、変調前と変調後とで比較して再現性を見るもので、図15に測定法を示します。

〈図13〉 測定時のビデオ(コンポジット)信号

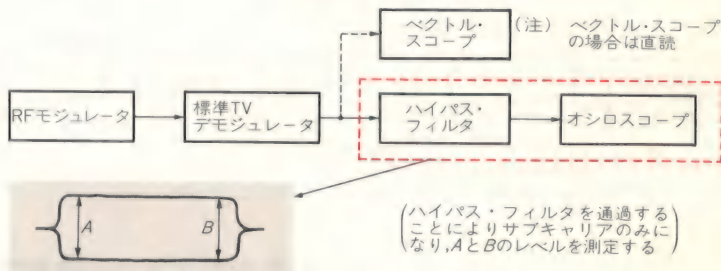


〈図14〉 水平同期信号の再現性の試験 (スペクトラム・アナライザ直統の場合)



A:変調後の水平同期信号

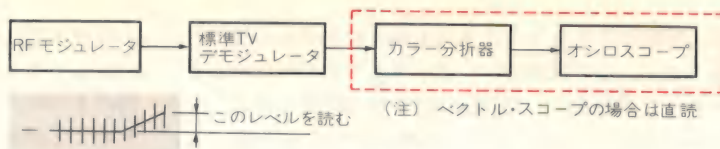
〈図15〉 微分利得の測定



$$DG = (1 - B/A) \times 100(\%) \text{ または } DG = 20 \log A/B(\text{dB})$$



〈図16〉 微分位相の測定



(3)の**微分位相**は、それぞれの変調されたカラー用サブキャリアの位相割り当てが、標準に対してどれくらいずれているかを測定するもので、**画像の色ずれの品質を決定する要素**となります。図16にその測定法を示します。

### ● 音声のFM送信上の注意点

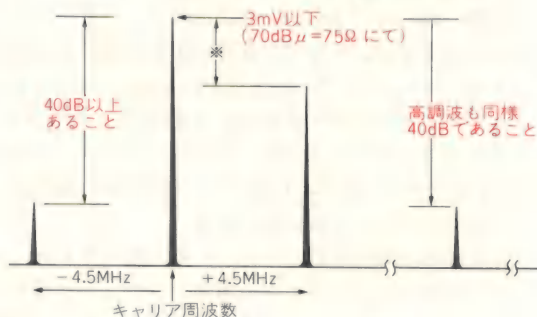
音声信号の送信については、FM通信の基本と共通の考えになりますので、ここではテレビRFモジュレータとしての注意点を示します。

日本国内におけるFM送信は、ワイヤレス・マイクなどでおなじみのトランジスタ1〜2個による構成が多いようですが、**米国の場合はFCC(Federal Communication Commission, 連邦通信委員会)規格(テレビ相互接続機器のキット)**がありますので注意が必要です。日本の場合も正式な製品などは、この規格に準拠するように設計されているようです。

図17にRF信号に関するFCC規格の概略を示します。ただし、図中の※印の部分のレベル差については、筆者の入手した資料の中では見い出すことができませんでした。

ただし、設定のポイントとしては、(キャリア+4.5

〈図17〉 FCCのRF信号規定



MHz)の4.5MHzと、サブキャリア用3.58MHzの差である920kHzが、テレビ受像機の画像ノイズになる可能性があるため、一般的には図17の※印の部分は、12〜16dBくらいのレベル差をもたせたほうがよいと思われる。

### ● RFモジュレータ回路の実例

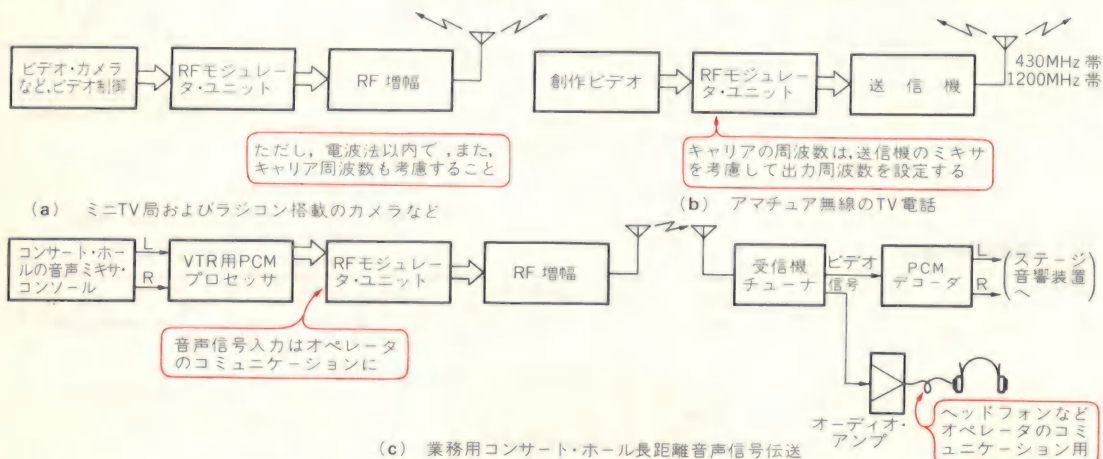
図18にここで使用する1チップRFモジュレータSN76874Nの構成を示します。また、図19に、SN76874Nを使った実際のテレビRFモジュレータの例を示します。ここで使うSN76874Nは、基本的には図11

## テレビRFモジュレータの利用できるところ

高性能のテレビRFモジュレータが実現できることより、発展した使用方法を図Aに示します。どれも空間に電波として送出する方法のため、電波法に

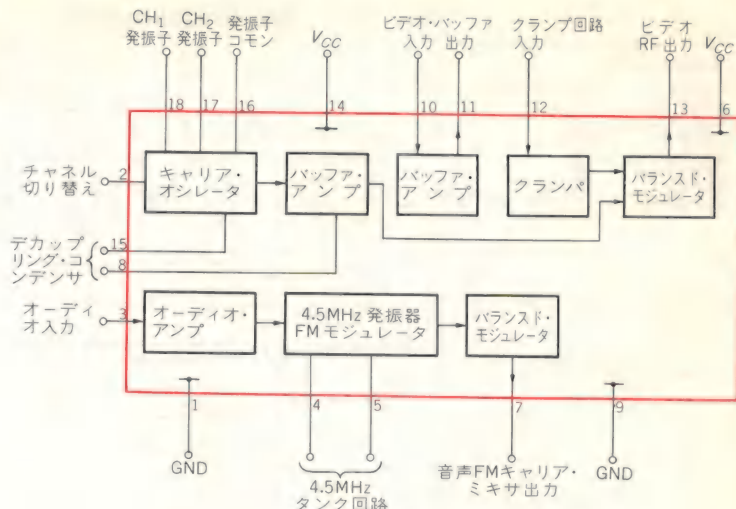
基づいて活用しなければならないことはいうまでもありませんが、いろいろと工夫してみるとおもしろいと思います。

〈図A〉 RFモジュレータの発展した利用例





〈図18〉 RFモジュレータ SN76874Nの構成



と同じ構成ですが、内部的にはバッファ・アンプなどが付加されて、ビデオ入力のインピーダンス・マッチングをやりやすくしています(ユーザにより75Ωや1kΩの違いがあるため)。

また、モニタ・テレビの電源がONの状態のときRFモジュレータ搭載機器の電源をONにすると、CRT画面上で画面があばれたりしますので、このへんについても改善が図られています。これは、RFモジュレータの電源立ち上がり時の一瞬に信号レベルが不安定になるため、これらを解消するために、電源立ち上がり時の時定数コントロール回路も内蔵されています。

図19において、キャリア用OSC回路にはSAW

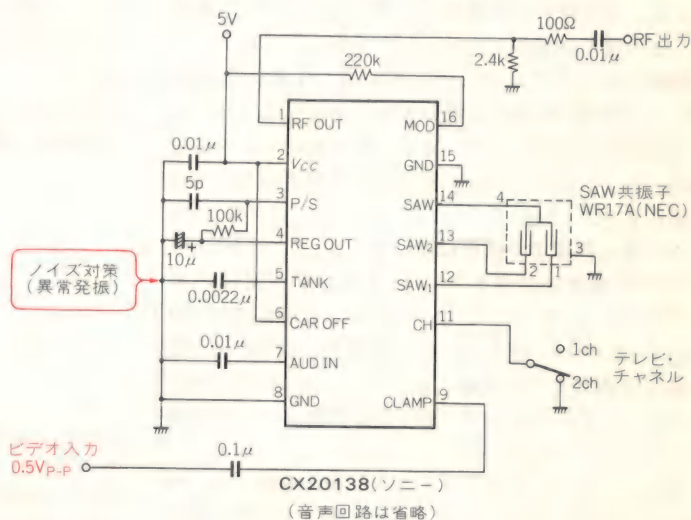
(Surface Acoustic Wave device: 表面弾性波)共振子を使用していますが、SAW共振子は、X'talのオバートーン共振に対して、キャリア周波数そのものを基本周波数として発振してくれますので高調波がきわめて少なく、後段でのフィルタ処理を簡略化することができます。

周辺回路のフィルタは、村田製作所のハイブリッド化したフィルタを使用することにより、図20でわかるように、チャンネル切り替えの際の周波数特性の切り替えが必要なくなりました。

図19の回路で220Ωと330Ωは、SN76874NのRF出力レベルに合わせて設定しています。回路図からわかる

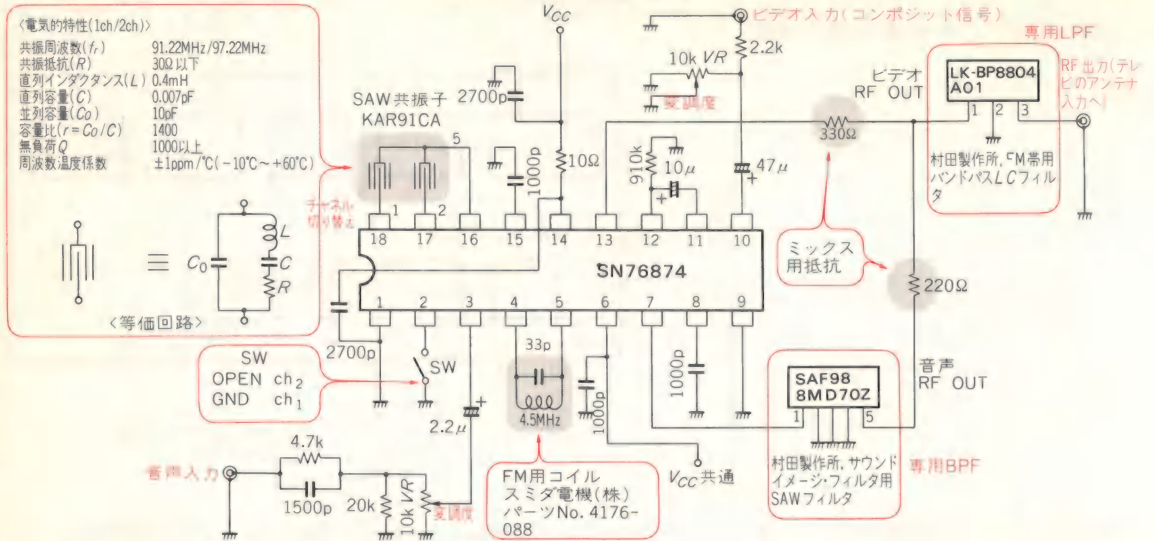
なお、RFモジュレータ用ICにはいろいろなものが発表されています。ごく簡単な回路構成で済ませたいという向きには、図Bのような方法もあります。この構成ではバッファは入っていませんし、音声回路もありませんので、かなりコンパクト化することができます。

〈図B〉 CX20138によるRFモジュレータ



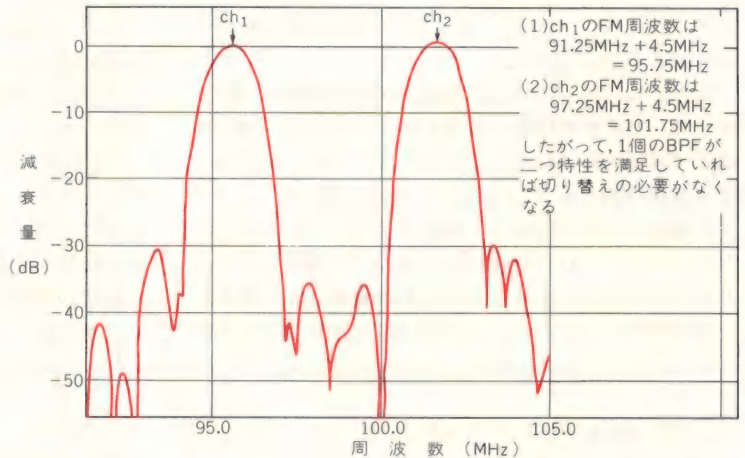


〈図19〉 SN76874Nの応用回路



〈図20〉<sup>(4)</sup>

FM用バンドパス・フィルタの周波数特性



ように、調整箇所は変調度と4.5MHzとコイルのみです。そのほかはすべて固定となります。

調整方法は、テレビのチャンネルを1chまたは2chにして、画像の色調が良好のポイントになるようにビデオ変調度を設定します。また、4.5MHzコイルの設定は、テレビのスピーカを聞きながら最適のところへ調整します。

### ● 5V単一電源での使用について

TIの半導体技術資料においては、電源電圧が6～7Vとなっていますので、パソコンでEIA-RS-232Cのポートがある場合は、その電源の±12Vの(+)側が使用可能です。これは、IC内部に5.6Vのレギュレータが入っているからで、**数十Ωの抵抗を介して+12Vへ接続**するようにします。

それ以外は、一般のセットは5V単一電源です。その点をTIに問い合わせたところ、一応4V付近から動作可能とのことでした。ICの電源規格が6～7Vという

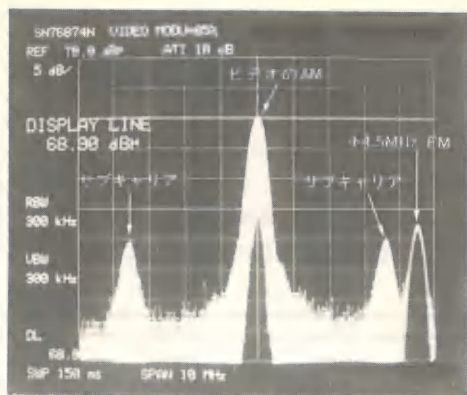
のは、内部に5.6Vのレギュレータ内蔵のためであり、5.6V付近をさけてそれ以下で使用する場合は、電圧値の変更ごとにビデオの変調度を再設定することにより、動作可能だとのメーカーの説明です。図19のボードによる各特性を写真2に示します。

〈渡辺成治〉

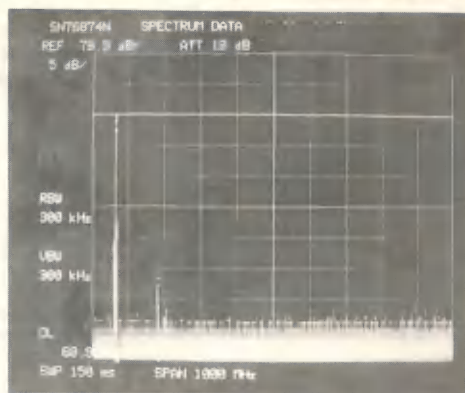
### ● 参考・引用・文献 ●

- (1) TI, 半導体技術資料No.20.
- (2) 京セラ㈱, SAW共振子データ・シート.
- (3) スミタ電機㈱, 4.5MHz FMコイル・データ・シート.
- (4)\* 村田製作所, RFコンパタ用フィルタ技術資料.
- (5) FCC規格(TV相互接続機器のキット).

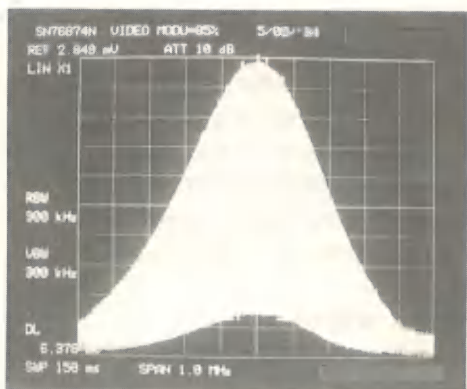




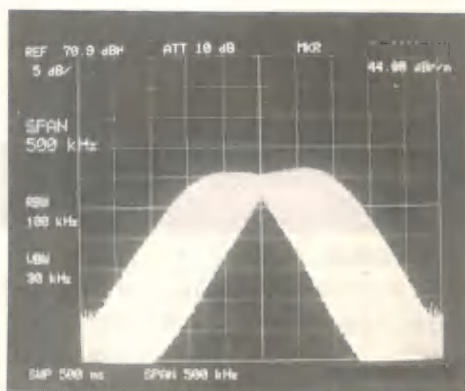
(a) ビデオ出力



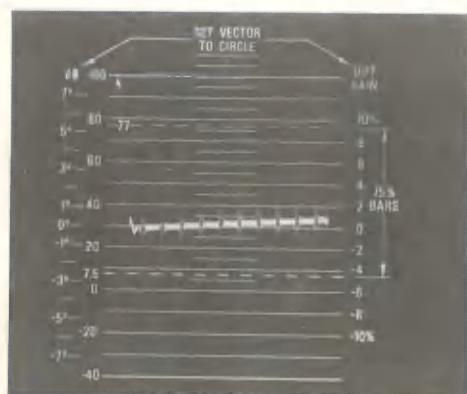
(b) 基本波と高次高調波スペクトラム



(c) ビデオ信号のAM(変調度85%)



(d) 音声信号のFM



(e) 微分利得  $DG = 1\%$



(f) 微分位相  $DP = 2.1\%$

〈写真2〉各部の特性



## ビデオ・エンハンサの設計・製作



藤崎 功  
峯岸英雄

画像をハードにしたりソフトにしたりするときに使用するのがエンハンサです。これは輝度信号の周波数特性を補正することにより実現できますが、ここでは専用ICを使用した効果的な手法を紹介します。

最近ビデオ映像の画質改善用として、ビデオ・エンハンサと呼ばれるものが使用されるようになり、各社よりユニットの発売、VTRやテレビ・カメラへの組み込みなど、普及の度を増しています。そこで、このビデオ・エンハンサの原理と、ビデオ・エンハンサ用IC NJM2209〔新日本無線㈱〕の使用方法を説明します。

### エンハンサの原理

#### ● 画像と輝度の関係

エンハンサの目的は画面をEnhance、すなわち強調することです。標準画面に対し、よりくっきり、またはソフトにコントロールすることです。しかし実際には、テープ録画や編集の際の画質劣化を防止する目的に使用される例が多いようです。

標準画面に対してソフトな画面、よりくっきりした画面とは、ビデオ信号で見るとどこが異なるのでし

うか。写真1および図1に標準画面とソフトな画面での輝度信号の波形を示します。信号の境界面の輝度信号差が大きければ、すなわち電圧差が大きければ、画面の境界がくっきりし、ハードな画面となります。ですから、標準画面の境界にピークをもたせ、写真2および図2のような波形にすることにより、よりくっきりした画面が作り出せることになります。

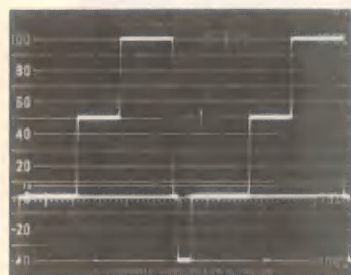
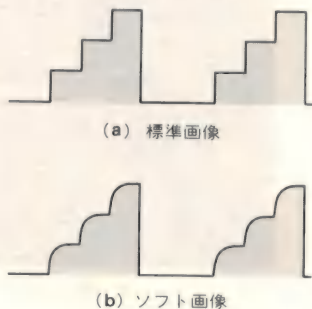
このように、標準輝度信号波形よりソフト→ハード（ピーキング）まで、輝度信号波形のエッジなどをコントロールする仕掛けを、一般にエンハンサと呼んでいます。

#### ● 電氣的に実現するには

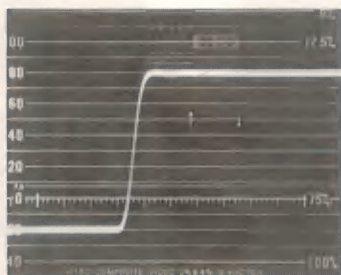
ではどうすればこのような効果を電氣的に実現できるのでしょうか。

輝度信号の周波数特性で考えると次のようになります。通常のビデオ信号（コンポジット映像信号）は約4MHzの帯域をもっています。すると輝度信号のエッ

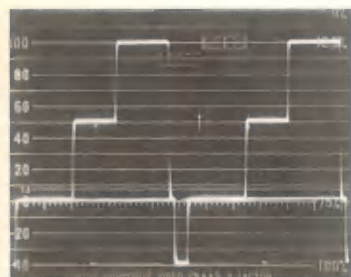
〈図1〉輝度信号の変化



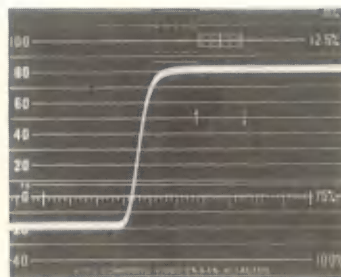
(a) 標準画面



(b) 標準画面の拡大



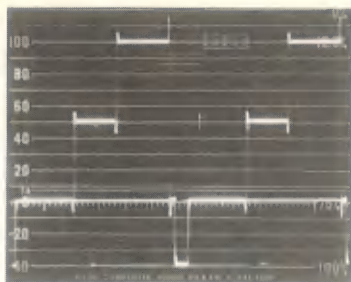
(c) ソフト画面



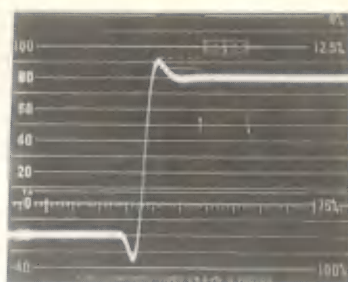
(d) ソフト画面の拡大

〈写真1〉エンハンスによる輝度信号の変化





(a) ハード画面



(b) ハード画面の拡大

〈写真2〉ハードな画像の輝度信号

ジ部分は2～3MHzの帯域に相当しますので、この部分を図3のようにカットすればソフトな画面に、強調すればくっきりしたハードな画面になります。

要するに、輝度信号増幅部の周波数特性をコントロールすればよいことになります。しかし、約4MHzの帯域をフラット、かつ位相遅れなく増幅することさえ容易ではないのですから、アンプの周波数特性を自由にコントロールすることはそう簡単にはできません。

## 画質補正IC NJM2209

NJM2209は、前述のようなエンハンス効果を得るために作られた画質補正用のICです。NJM2209では、ビデオ帯域内の周波数特性を制御するのではなく、輝度信号を微分して、微分出力を原信号に加算する手法で同様の効果を得ています。図4にNJM2209の構成を示します。表1が電気的特性です。

### ● 基本的な動作原理

図5にNJM2209を使った基本回路を示します。このICはシングル・インラインの9ピン・パッケージで、5V単一電源で動作します。

信号の流れにそって動作を説明しますと、まず1番ピンに1V<sub>P-P</sub>の輝度信号が入力されます。入力信号としてはコンポジット信号からY/C分離を行い、3.58MHzの色成分をカットしておく必要があります。信

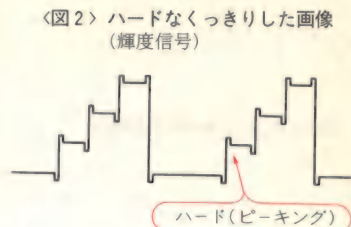
号は直ちにクランプ(直流再生)され、次にエンハンスに不要な同期分をスライスして除去します。そして輝度成分だけになった後、波形の変化分(輪郭部)を取り出せばよいわけです。

実際には図5の中の微分コントロール・アンプ部で2次微分波形を取り出します。そして取り出した波形を元の輝度信号と合成すれば、エンハンスされた信号となります。

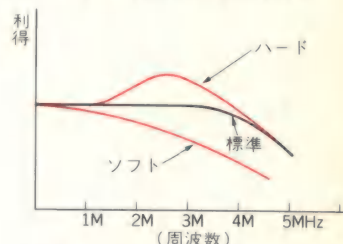
この2次微分波形の微分電圧量は、9番ピンの外部DC電圧で設定することにより、自由に可変することができます。例えば、微分波形の振幅が大きなハードな画面にしたい場合は9番ピン電圧を3V、なめらかでソフトな画面にしたい場合は0.45Vというように、0.45～3Vの間で自由に画像の質を可変できます。

なお9番ピンをグラウンドに落とすことにより、波形操作を行わない入力スルー信号が6番ピンより出力されます。9番ピンの外部回路はDC電圧の制御回路です。1番、2番ピンの外部回路により2次微分波形を作るわけですが、ここのLCRの値はエンハンスをかけたい周波数に合わせて選んでください。

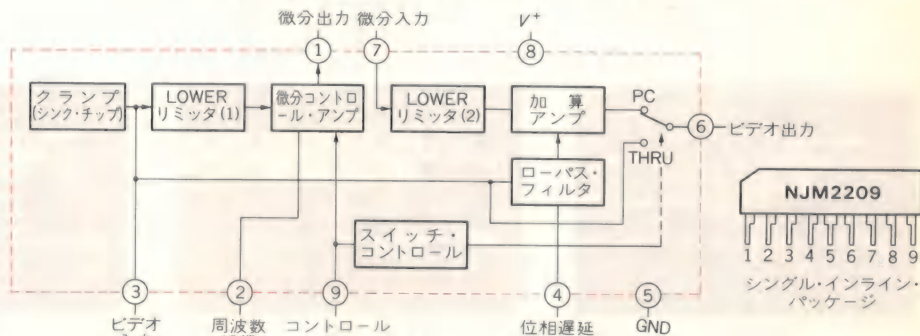
図5の外部定数は、ハード画質時に2M～3MHz



〈図3〉輝度信号の周波数特性



〈図4〉  
NJM2209の構成  
〔新日本無線㈱〕



(a) ブロック図

(b) ピン接続

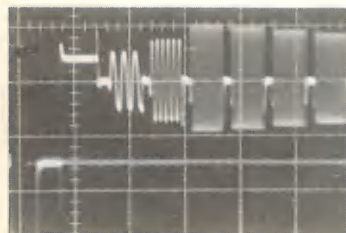
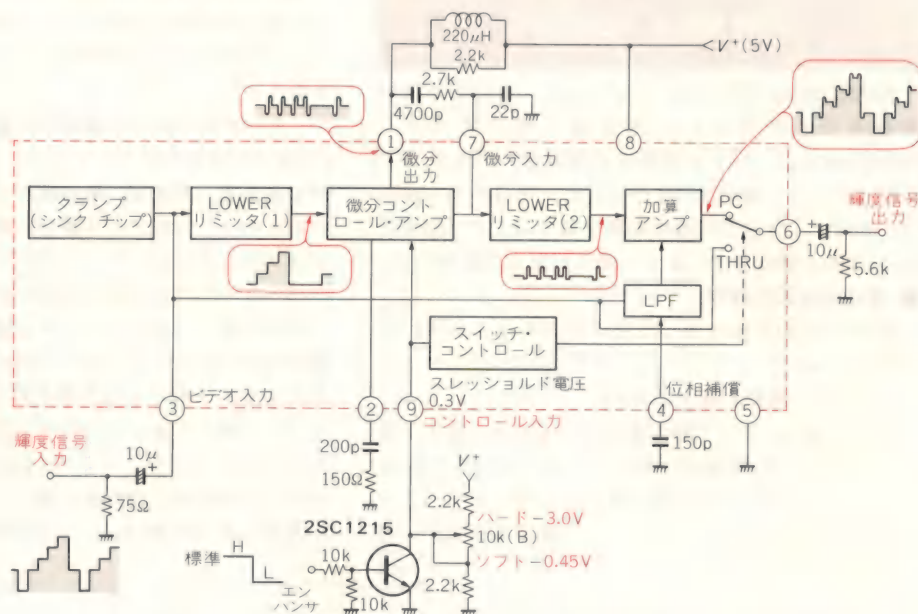


〈表1〉 NJM2209の電気的特性

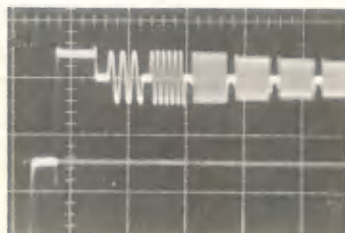
項 目		測 定 条 件	最小	標準	最大	単位
消費電流		入力無信号時	—	7.5	10	mA
リミッタ・レベル(1)		SYNCレベル>0.35V, ビデオ信号を入力	0.23	0.27	0.31	V
リミッタ・レベル(2)		$f=100\text{kHz}$ , $1\text{V}_{\text{P-P}}$ 正弦波入力	0.21	0.25	0.29	V
コントロール・アンプ利得	H	$f=100\text{kHz}$ , $0.1\text{V}_{\text{rms}}$ 正弦波入力	-2	-0.9	0	dB
	M		-12	-10	-8	dB
	L	$G=20\log_{10}\frac{V_{\text{OUT}}}{V_{\text{IN}}}$ (dB)	—	—	-28	dB
加算アンプ利得	⑦ピン入力	$f=100\text{kHz}$ , $200\text{mV}_{\text{P-P}}$ 正弦波入力 $G=20\log_{10}(V_{\text{OUT}}/V_{\text{IN}})$ (dB)	-1.6	-0.6	0.4	dB
	③ピン入力	$1\text{V}_{\text{P-P}}$ ビデオ信号入力 $G=20\log_{10}(V_{\text{OUT}}/V_{\text{IN}})$ (dB)	-1	0	+1	dB
スイッチ・クロストーク		$f=2\text{MHz}$ , $1\text{V}_{\text{P-P}}$ 正弦波入力 $C_{\text{SW}}=20\log_{10}V(0\text{V})/V(2.8\text{V})$ (dB)	—	-50	—	dB
スルー時の利得		$1\text{V}_{\text{P-P}}$ ビデオ信号入力 $G_T=20\log_{10}(V_{\text{OUT}}/V_{\text{IN}})$ (dB)	-1	0	1	dB
スイッチ・コントロール・スレッシュホールド電圧		$f=100\text{kHz}$ , $1\text{V}_{\text{P-P}}$ 正弦波入力 $-40\text{dB}=20\log_{10}(V_{\text{OUT}}/V_{\text{IN}})$	0.2	0.3	0.4	V
微分利得	画質補正時	$DG, DP$ 測定メータ	—	1	3	%
	スルー時	ビデオ信号 $1\text{V}_{\text{P-P}}$ (ステア・ステップ)	—	0	3	%
⑥ピン電圧	画質補正時		—	1.8	—	V
	スルー時		—	2.0	—	V

(注)  $V^+=5\text{V}_{\text{DC}}$ ,  $T_a=25^\circ\text{C}$ , 標準測定回路での値

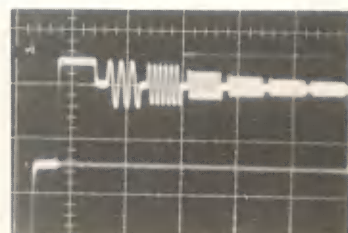
〈図5〉  
基本的な応用回路



(a) ハード



(b) 標準



(c) ソフト

〈写真3〉 マルチ・バースト信号によるエンハンスされた波形



を約2倍にし、ソフト画質時に2M~3MHzを約0.4倍にしています。写真3に100kHz~10MHzの周波数成分をもったマルチ・パースト信号を入力した時の入力波形、ハード、ソフトの各波形を示します。

この図5の例では、1番ピンで微分波形を得た後に4700pF、2.7k $\Omega$ 、22pFで波形を整え、7番ピンに入力しています。そしてLOWERリミッタ(2)で波形の下側をスライスします。これは微分波形の下側が同期成分に入りこむを防ぐためです。

波形操作はここまでで、次の加算アンプ部で微分波形と入力波形を加え合わせ最終波形にします。4番ピンの外付け容量は、エンハンス波形がきれいになるよう選びます。メーカーの推奨値は150pFです。

ビデオ信号(輝度信号)はIC内部のビデオ・スイッチを通った後、6番ピンに出力されます。6番ピンの負荷としては2~3k $\Omega$ 以上にします。この後に外部回路で色信号とミックスしてコンポジット信号にします。ミックス回路の出力は1V<sub>P-P</sub>に選定するのがよいでしょう。また輝度成分、色成分もボリュームで自由に設定できるようにするとよいでしょう。

なお、NJM2209はテレビ入力端子の75 $\Omega$ を直接ドライブできませんので、後段にビデオ・バッファが必要です。

## ● 実際のエンハンサ

図6が実際のエンハンサの回路例です。NJM2209は輝度信号を操作してエンハンス効果を得る方式のため、NTSCコンポジット信号を直接に入力することはできません。

そこで第1章で述べたような方法で、コンポジット信号をフィルタを使って輝度信号と色信号に分離します。輝度信号は3.58MHzの色副搬送周波トラップを通すことにより得られます。また、色信号は3.58MHzのバンドパス・フィルタにより分離します。

図7に両フィルタの周波数特性を示します。写真4はコンポジット信号と分離された色信号、輝度信号の波形です。この輝度信号を、NJM2209の3番ピンに入力します。なお、画像の質をより的確にしたい場合は、くし形フィルタによるY/C分離のほうが優れています。

そして、前に述べた動作説明に従い操作された輝度信号は6番ピンに出力されます。この信号に先ほど分離した色信号を加算し、エンハンスされたコンポジット信号とします。

このエンハンサ回路では、入力をハイ・インピーダンスで受けていますので、回路中のビデオ信号は2V<sub>P-P</sub>となっています。出力バッファは、無負荷で2

## イメージ・プリプロセッサ

ビデオ・エンハンサは、VTRやカメラとテレビとの間に置いて、画質のハード/ソフトを操作するものですが、産業用の本格的な画像処理システムでも入力画像の質が画像処理に大きな影響を与えます。そこで、画像処理システムの中でも、エンハンサと似た感じの働きをする画像前処理の技術が注目されており、イメージ・プリプロセッサという呼び方がされています。

これは、イメージを捉えるカメラと画像入力ボードとの間に置かれるもので、代表的なものとしてはコントラスト・コンバータ、マルチシェーディング・ユニットなどと呼ばれるものです。また、第2章、第3章で紹介したNTSCデコーダ、NTSCエンコーダも、イメージ・プリプロセッサの一種といえます。

### ●コントラスト・コンバータ

コントラストの非常に弱い映像、例えば色むらや汚れ、暗い照明下での映像のコントラストを強調し、観測しやすい画像に変換するものです。

これは、映像信号の中の輝度信号にスレッシュホル

ド・レベルとクリップ・レベルとを別々に設定できるようにして、任意の輝度部分のみを拡大できるようにしたものです。

### ●マルチシェーディング・ユニット

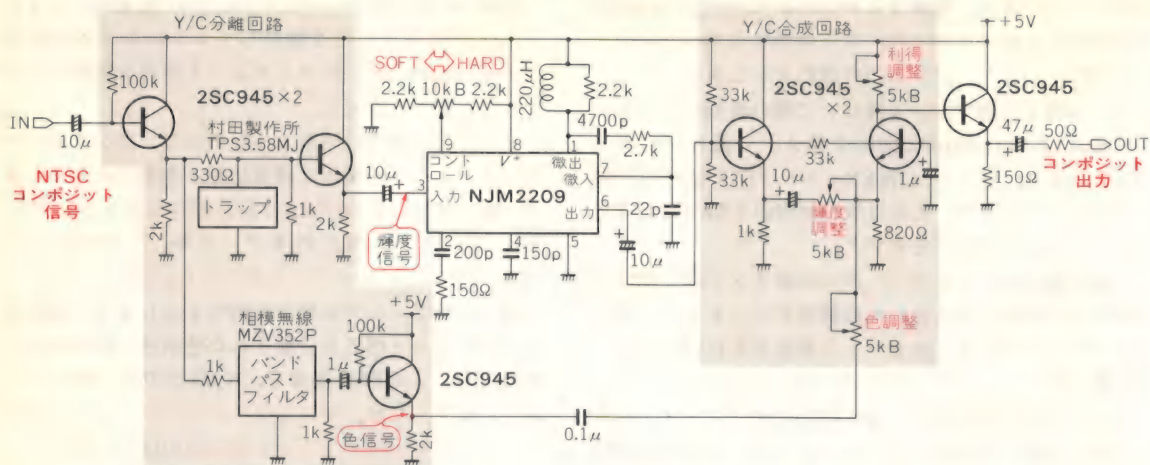
これは、画像を捉えるときに照明状態やカメラ特性の影響から生じる照度むらを補正し、均一なバック・グラウンドを得るためのものです。これにはイメージ・メモリを使い、シェーディング(輝度むら)を補正するための画像を逆加算するような手法が使われています。



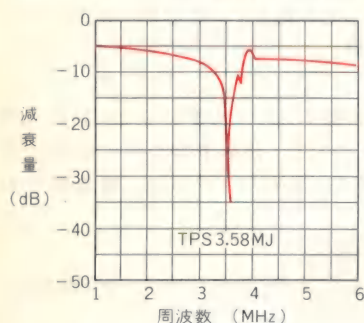
〈写真A〉 イメージ・プリプロセッサの一例  
〔株〕サンコム・テクニカル



〈図6〉エンハンサ回路

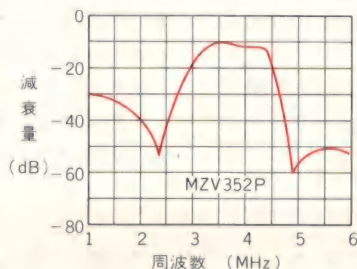


〈図7〉Y/C分離用フィルタの特性



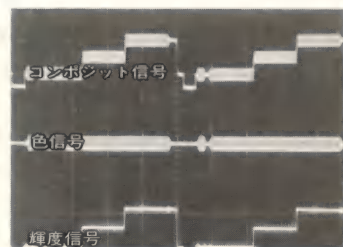
駆動インピーダンス 100Ω  
終端インピーダンス 1kΩ  
トラップ減衰量 27dB(typ)

(a) 輝度信号用色副搬送周波トラップ(村田製作所)



入出力インピーダンス 1.0kΩ  
しゃ断周波数 2.90, 4.15MHz  
減衰量 20.0±5dB @ 2MHz  
30.0dB(min) @ 4.86MHz  
遅延時間 0.52μs

(b) 色信号用バンドパス・フィルタ(相模無線製作所)



〈写真4〉コンポジット信号のY/C分離

$V_{P-P}$ , 75Ω負荷で1V<sub>P-P</sub>となるように利得調整をしてください。

NJM2209を使うと、このような簡単な回路でエンハンス効果を楽しむことができます。なお、図6の回路は入力コンポジット信号としたために輝度、色信号分離が必要で、若干めんどろなフィルタが必要でしたが、実際のVTRやテレビ・カメラの内部では、コン

ポジット前の輝度信号と色信号があるはずで、この信号を探し出せば、回路をより簡単にすることができます。

#### ●引用文献●

- (1) NJM2209, 新日本無線 86半導体データ・ブック, パイポーラIC編, (発売)CQ出版社,





## NTSCデコード&スーパーインポーズの設計・製作

宏 祐二  
松井俊也

映像の中にパソコンなどからの文字を挿入することをスーパーインポーズといいます。ここではその一般的な手法と、RGB信号へスーパーするようにした具体例とを紹介します。

最近、ニュー・メディア対応を意識して、テレビ自体にNTSCコンポジット信号の入出力端子を設けたり、RGB信号で直接カラー信号を受け付けるディスプレイが普及してきました。そして、パソコンなどとの融合をめざしたシステムが考えられつつあります。

また、映像信号をパソコンなどに取り込んで画像処理を行うという例が増えてきています。

そこでNTSC信号をいかにして取り扱うかという技術が問題となりますが、一般にはNTSC信号からR・G・Bのそれぞれの色信号に分離できれば、その後の処理は非常にやりやすくなります。画像処理用の信号とするにしても、パソコンからの信号をスーパーインポーズするにしても、この技術は非常に役立ちます。

この章では、以上のような理由からNTSCコンポジット信号のRGB信号(コンポーネント信号という)への変換と、パソコンなどからのスーパーインポーズの方法について紹介します。

### スーパーインポーズとは

外部信号を映像信号に重畳することをスーパーインポーズと呼んでいます。まず、このスーパーインポーズの手法について、説明を行います。

テレビなどを見ていると、よく時刻や字幕スーパーのように、実際の映像信号に対して別の信号を重ね合わせているのに気付くことがあると思います。

このように外部(テレビ以外の機器)からの情報をテレビの映像信号に重ね合わせることをスーパーインポーズと呼んでいます。

映像信号をスーパーインポーズする場合、入力される信号の種類や表示するCRTの種類、および表示方法によって必要とする処理回路が異なります。ここではまず、各種の組み合わせについて一般論を解説します。

#### ● 水平、垂直同期の問題

いずれの組み合わせにおいても必要不可欠なのは、テレビ画面上での水平、垂直同期を合わせるという問題です。

加算しようとする二つの映像信号はたとえ同期周波

数が等しくても、これらが時間的に一致していなければ画像として表現することはできません。そこで、どちらか一方のクロックを基準として片方のクロックを同期させる工夫が必要となります。ここでは、テレビ画面の中にパソコンやマイコンなどの画像、文字などをスーパーすることを前提にしてそれぞれ、どんな方法があるかを考えてみましょう。

マイコンなどでの一般的な文字発生メカニズムは、図1のような構成になっています。CPUのクロックと、文字などを生成して画像化するためのCRTコントローラのクロックはまったく別世界で動いています。したがって、これらを同期させる必要はありませんが、CRTコントローラのクロックは、スーパーされる側の入力映像信号に同期させなければなりません。

この具体的な方法としてはPLL(Phase Locked Loop; 位相同期回路)による同期化が一般的です。CRTコントローラのマスタ・クロックは、通常水平周波数の $N$ 倍になっていますので、この周波数をVCO(電圧制御発振器)で発振させ、お互い(入力の映像信号とスーパーするCRTコントローラ)の水平同期信号同士を位相比較してPLLにかけます。

#### ● 同期分離の問題

入力映像信号がRGBのコンポーネント信号であれば、同期信号は別系統で出力されているため、これをそのまま使用することができます。しかし、コンポジット映像信号(NTSC信号)の場合には、この中に含まれる同期信号だけを取り出す必要があります。

同期分離の原理は、映像信号の同期部分だけを強調増幅し、フィルタで水平同期と垂直同期に分けるという方法です。テレビ用のICで便利なものがありますので、それを活用すればよいでしょう(第6章参照)。

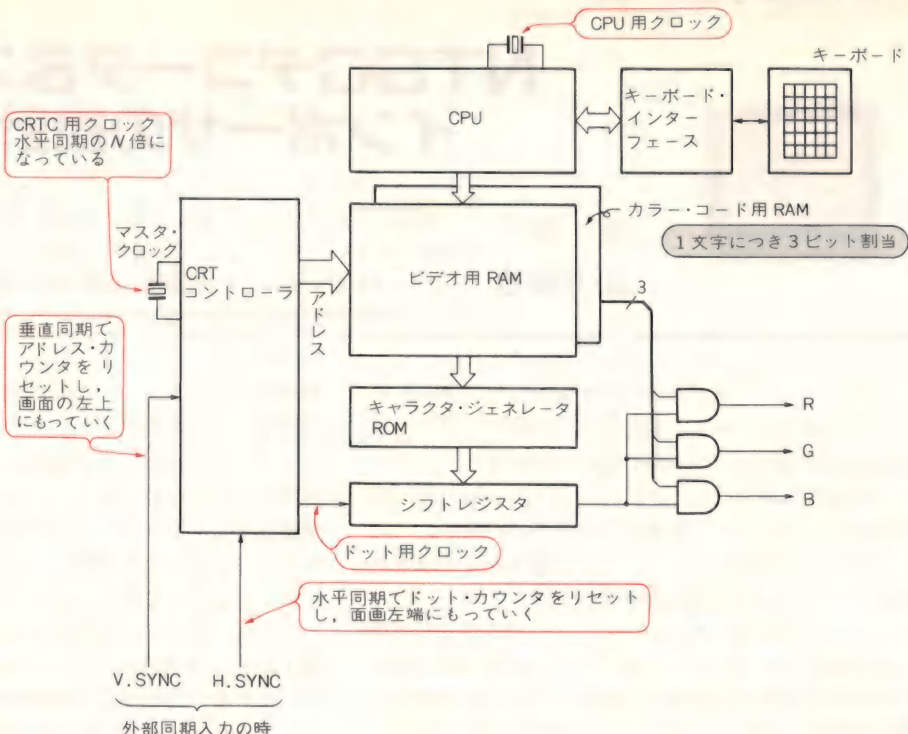
#### ● カラー・バースト同期について

一般の家庭用テレビの信号に、パソコンなどの文字信号をスーパーインポーズして、家庭用テレビに写して見ようとした場合には、さらに複雑となります。すなわち、入力信号も出力信号もコンポジット映像信号であるということです。

この場合は、カラー・バースト信号の3.58MHzも同



〈図1〉  
一般的な文字発生部の  
ブロック図



期をかける必要があります。水平同期の場合と同様の方法で、コンポジット信号の中からカラー・バースト信号を取り出し、CRTコントローラ側の3.58MHzとPLLにかけます。

最近はコンポジット映像信号同士でスーパーインポーズできるICが発表されていますが、具体的な方法は第6章で詳しく紹介しています。

これらのICが使用できない場合は、またまた複雑なことを行わなければなりません。すなわち、コンポジット信号をいったんRGB信号まで分解(デコード)し、パソコンのRGB信号と合成した後に、再びコンポジット信号に変調(エンコード)するという作業を行います。しかも、これらの作業はアナログ的に行わなくてはならないのでちょっと大変です。

現実的には文字に色を付けなければ(白文字のみにする)、バースト同期の必要はなく、単純にコンポジット信号レベルでのスーパーインポーズが可能です。

#### ● モニタ・テレビ側の問題

一般の家庭用テレビの場合、水平方向の解像度はパソコン用のCRTに比べて悪くなっています。そのために、せっかくスーパーインポーズしても、小さい文字が出てこなかったり、ボケたりします。したがって、CRTに合わせて文字を大きくするか、もしくはパソコン用のCRTを使用する必要があります。

この問題を解決するために、文字のふちの部分強調したり、映像回路の高域補正をしたりするテクニッ

クがあります。

また、パソコン用のCRTを使用しても、RGB入力がアナログRGB対応になっていないと、アナログ画面で表現することができません。

たとえ水平、垂直の同期が同一規格であっても、このように解像度に問題があると困ってしまいます。そこで、前もってCRTの解像度を調べておくことが必要です。最近は家庭用テレビでも高解像度のものが出まわっていますので、これらを利用するのもよいでしょう。

以上述べたように、たんにスーパーインポーズといっても、その方法は様々で、扱う信号の種類、機器に合わせて総合的に検討しなければなりません。

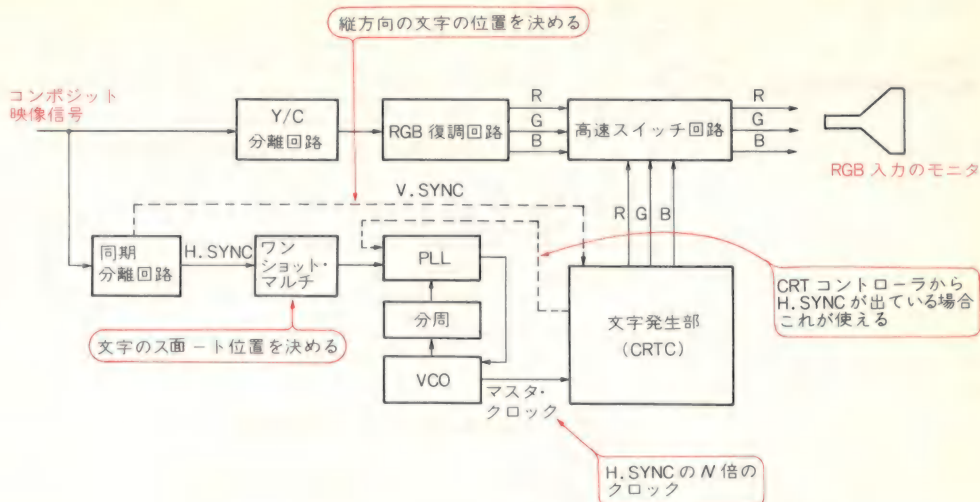
文字発生部分を自作する場合は、スーパーインポーズのことをあらかじめ考慮した設計にしておき、クロックや同期信号などを外部から入力できるようにしておくといよいでしょう。しかし、既製のセットを使うとこの点の配慮がなされていない場合があるため、意外に苦勞することがあります。

#### ● コンポジット信号にRGB信号をスーパーする

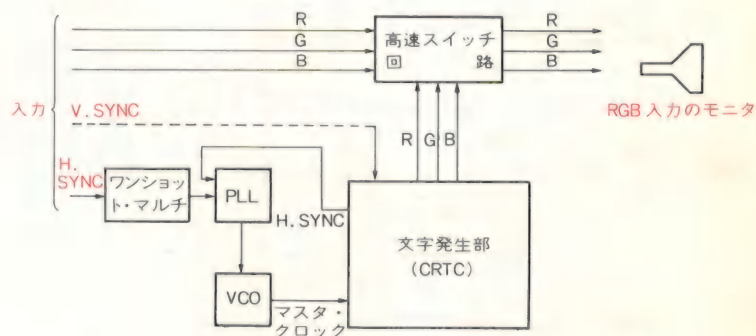
図2はコンポジット信号をRGB信号に分解(デコード)し、アナログ・スイッチを使って合成する例です。この場合、モニタはアナログRGB対応でなければなりません。V. SYNCをCRTCに与えると、文字を出す位置をいつでも特定することができて便利ですが、本章の後半ではこの方式のものを設計・製作することに



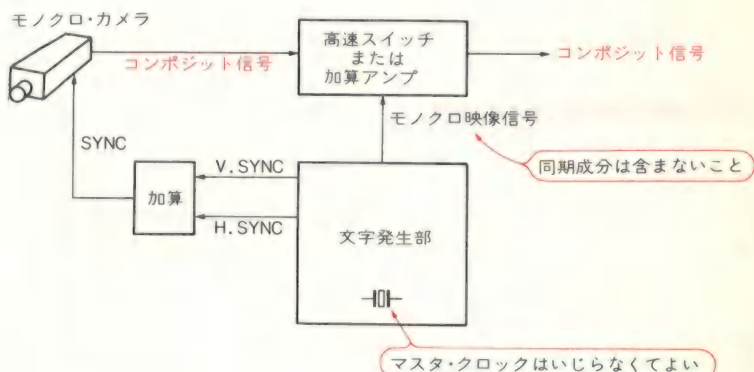
〈図2〉  
コンポジット映像信号にRGB信号を合成する場合



〈図3〉 RGB信号同士の合成



〈図4〉  
信号源の同期を逆に供給する方式  
(モノクロ・カメラの場合)



します。

なお、VCOのクロックをH. SYNCと同じ周波数まで分周して比較しますが、これは通常はCRTCから出ていることが多いようです。

図3は、RGB出力のAVテレビなどを合成するとき 사용합니다。V. SYNCとH. SYNCが合成されている場合は、簡単なハイパス・フィルタでH. SYNCのみを取り出すことができます。

#### ● カメラ側に同期信号を送る場合

図4は入力映像信号がカメラ側で作られる場合の例

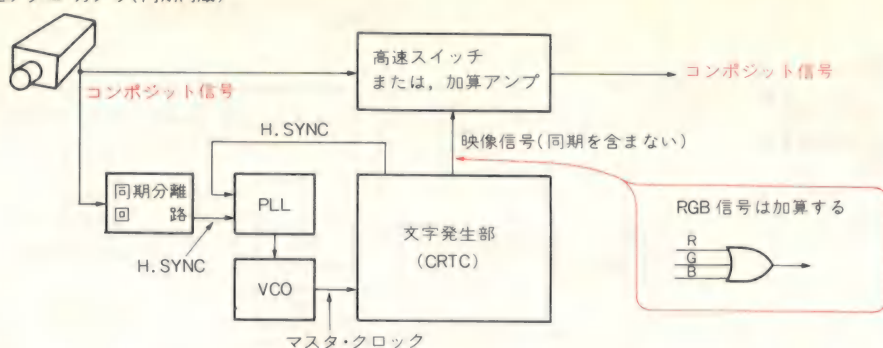
で、モノクロで十分な場合(最近あまりはやらないが)、またカラー・カメラは高価で使えない場合に役立つ方式です。

普通のテレビ・カメラは同期発生を内蔵していますので、むしろ図5の方式が現実的でしょう。合成するのに高速スイッチを使用する場合は問題ありませんが、加算アンプを使う場合は、文字信号に同期成分が含まれないように注意します。文字信号を反転すれば黒文字も出せます。

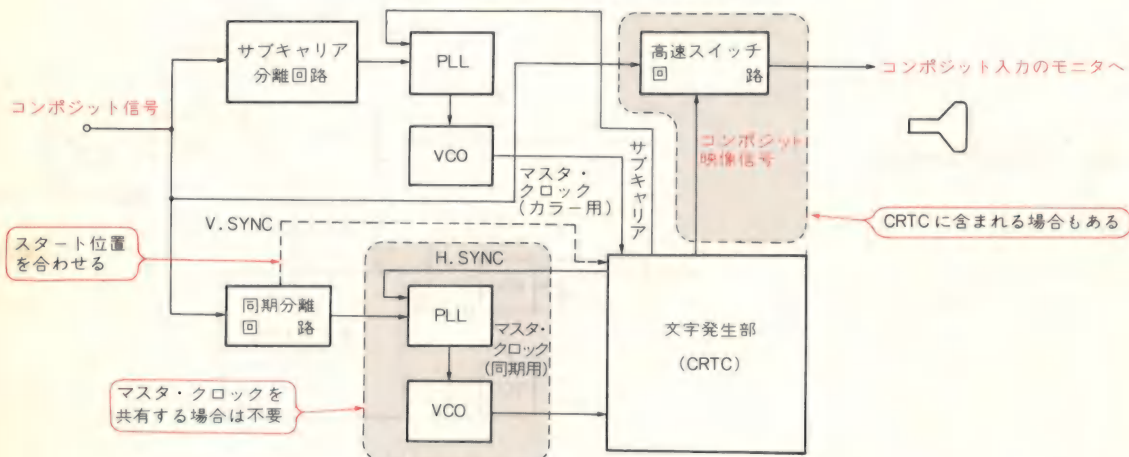
カラーのCRTCの場合は、特定の色だけ文字として



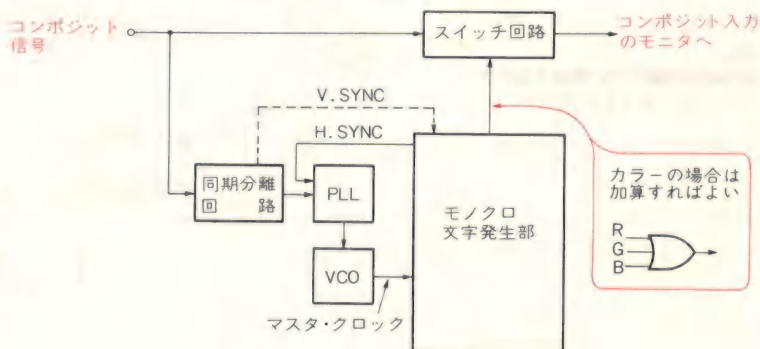
〈図5〉  
モノクロ信号同士の合成



〈図6〉 入出力共にコンポジット信号の場合



〈図7〉  
カラー信号に文字を白で合成する場合



出すことも可能です。応用はいろいろあります。

### ● コンポジット信号同士の合成

図6は入出力共にコンポジット信号である場合で、回路は複雑です。カラー・バースト信号(サブキャリア)は、1個のマスタ・クロックから作ることもありますが、サブキャリアは水平同期の $N$ 倍にはならないので、正確には同期が一致しませんが、目につくほどではありません。マスタ・クロックをサブキャリアの3倍ぐらいに選ぶとPLLが1個ですみます。

図7は、文字に色づけしないという条件付きの合成

です。コンポジット信号でもカラー・バーストが不要なため、文字の部分だけモノクロ信号になります。スイッチも1系統ですみます。モノクロのCRTCが使えるのでわりと実用的です。

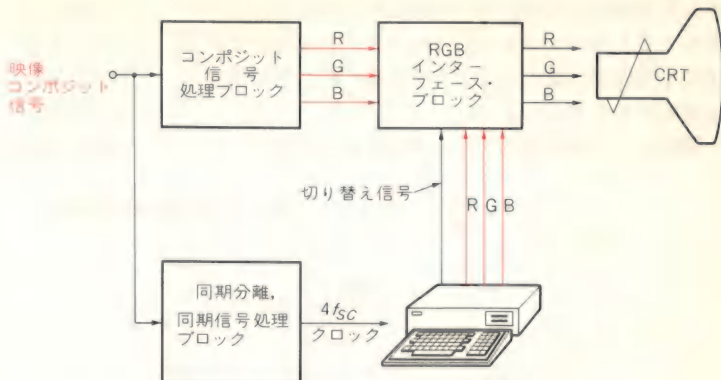
〈宏 祐二〉

## RGB信号へのスーパインポーズ

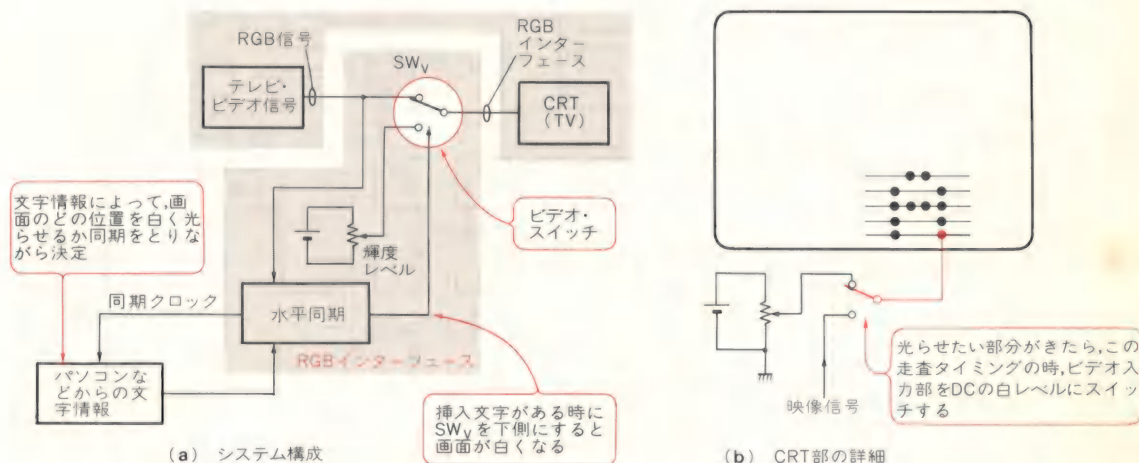
では、実際のスーパインポーズ回路について考えてみることにしましょう。ここでは、図8のような構成に基づく方法について設計することになります。



〈図8〉  
スーパインポーズ・システムのブロック図



〈図9〉スーパインポーズの原理



## ● 基本的なシステム構成

図9が模式的に書いたスーパインポーズの構成です。

まず、外部機器の信号の出力タイミングで、テレビ映像信号と外部機器信号を高速で切り替えます。タイミングのみを考えれば、この切り替え動作でスーパインポーズは実現できます。

しかし、実際にはDC(直流)レベル、AC(信号)レベルの相互のマッチングをとる必要があります。そのため、インターフェース部にはRGBインターフェース用のICを使用します。テレビとの接続にはEIAJ 21ピン・マルチ・コネクタを使用します。

スーパインポーズを大きくブロックに分けると、まず、映像コンポジット信号をRGBコンポーネント信号にするブロック、外部RGBと映像RGBコンポーネント信号を高速に切り替えるRGBインターフェース部、外部RGB信号と映像信号とで同期をとる同期ブロックとに分けられます。

## ● コンポジット信号の処理

この回路は、NTSCコンポジット信号をRGBコンポーネント信号に変換(復調)するブロックです。

実際の回路は最後の図18(50ページ)に示します。

基本的には、第2章で述べた輝度信号と色信号を分離するY/C分離回路、輝度信号処理回路、色信号処理回路から構成されています。この回路においては、NECの $\mu$ PC1352Cを用います。

この $\mu$ PC1352Cは、図10に示すように映像増幅および、画質やDC再生を行うビデオ信号処理回路、クロマ帯域増幅、色同期、色復調、カラー、色相調整を行うクロマ信号処理回路を含み、処理後の信号はRGB原色で出力されます。

## ● RGBインターフェース

コンポジット信号処理ブロックで復調されたRGB信号を、高速でパソコンなどの外部信号と切り替えるためのブロックです。 $\mu$ PC1397Cを使用しています。このICのブロックは、図11に示すとおりです。

端子(3)、(5)、(7)は映像コンポーネントRGB信号の入力端子で、端子(2)、(4)、(6)が外部RGBコンポーネント(パソコンなど)信号の入力端子です。端子(9)からの切り替え信号により、この二つの信号の切り替えを高速で行います。このタイミングを図12に示します。

## ● 同期分離と同期信号処理

同期信号処理ブロックでは、映像信号の同期信号と、

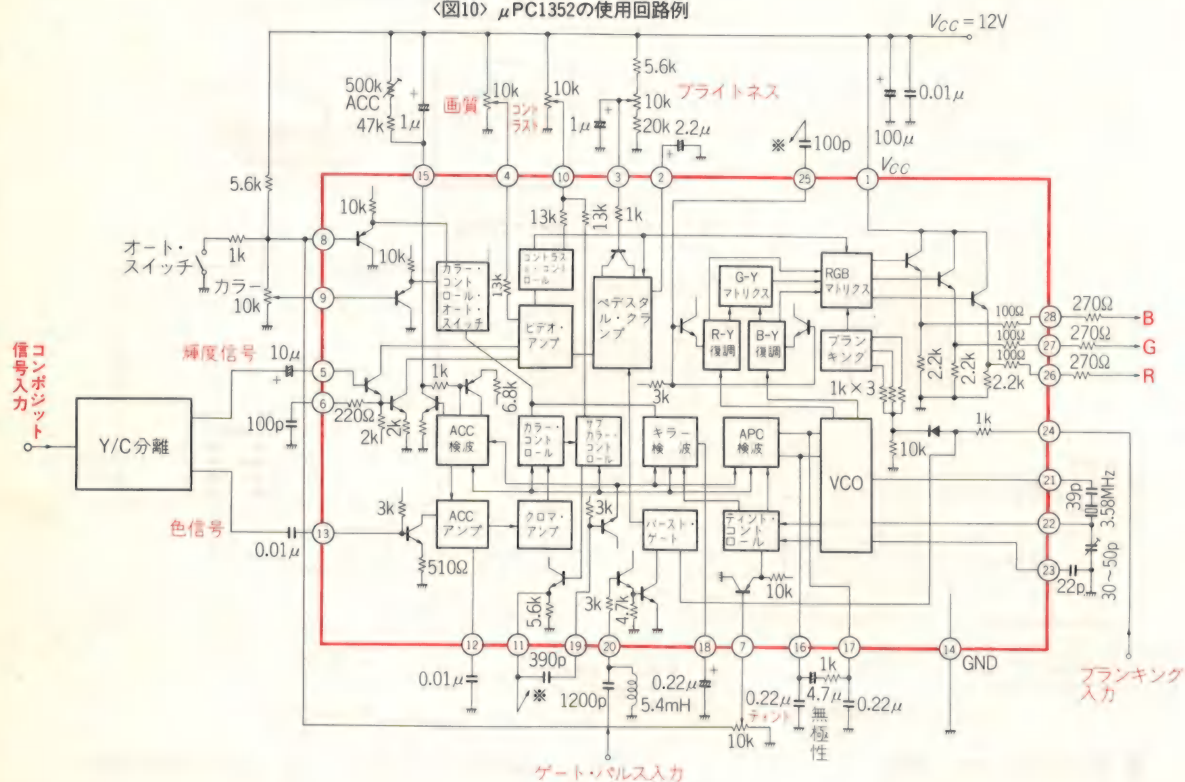


外部入力信号の同期信号の周波数および位相を同期させるを行っています。このとき、外部信号に映像信号を同期させるのではなく、映像信号に外部信号を同期させることにします。

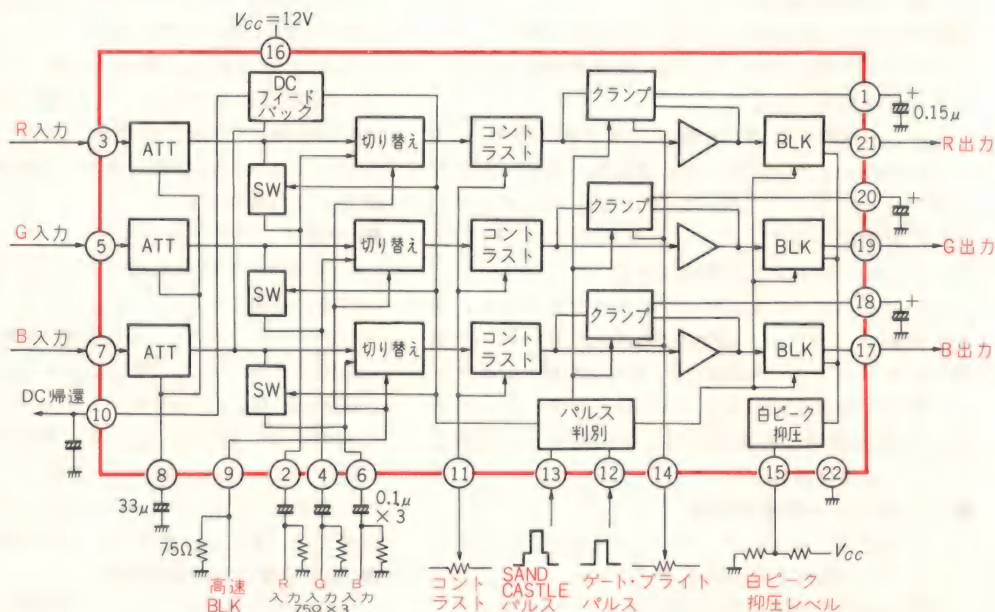
図13にその基本的なブロック図を示します。

原理的には、映像信号を同期分離して水平信号を取り出します。ただし、この同期分離信号には、垂直同期信号や等価パルスを含んでいるため、この信号をそのまま位相クロック用の信号として使用すると、VCOの発振信号の安定が良くありません。つまり、

〈図10〉  $\mu$ PC1352の使用回路例



〈図11〉  $\mu$ PC1397Cのブロック図





PLLのロック・レンジ(保持範囲), キャプチャ・レンジ(引き込み範囲)が狭くなってしまいます. そのため, 別にVCOをもう一つもたせ, **ロック・レンジ, キャプチャ・レンジを広げてやる必要があります.**

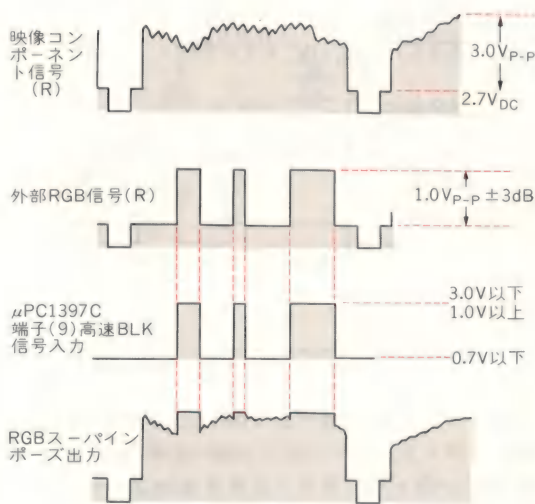
一つめのVCOは,  $f_H = 15.734\text{kHz}$ で発振しており, このVCOで, まず映像信号から同期分離した同期信号の水平信号に位相ロックした信号を作り出します. そして, このVCOの信号を少し位相をずらせることにより, 表示する位置をずらすことができますようにします. この一つめのVCOをVCO<sub>1</sub>とし,  $8f_{sc}$ で発振しているVCOをVCO<sub>2</sub>とします. そして, **VCO<sub>1</sub>の信**

号とVCO<sub>2</sub>を分周して作った信号 $f_H$ とで位相比較を行います. さらに, この位相差の検出力をローパス・フィルタでDCに変換し, このDC電圧でVCOの発振周波数を制御します. いわゆるPLL(フェーズ・ロック・ループ)回路です.

ここでは, 水平信号同期分離とVCO<sub>1</sub>, AFC(位相比較回路)にNECの $\mu\text{PC1377C}$ を使用しています. **このICは映像信号を同期分離し, その同期分離した水平同期信号のタイミングで位相比較を行い, 水平同期信号と位相の一致した発振出力を出力するICです.** このICのブロック図を図14に示します.

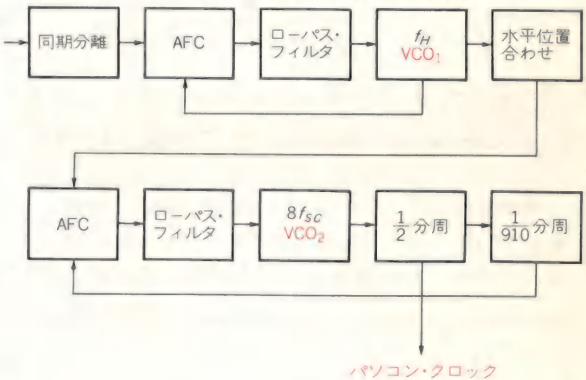
VCOは色副搬送波 $f_{sc}$ (3.5797MHz)の8倍の周波数, 28.63636MHzで発振しており, これを1/1820分周することで $f_H$ (水平同期周波数)=15.734kHzを作り出しています. そして, この $f_H$ に同期した **$4f_{sc}$ 信号14.31818MHzをパソコンのCRTクロックとして使用することにより, 映像(テレビ)信号に同期した外部信号を作ることができます.**

〈図12〉 スーパーバイポーズ・タイミング

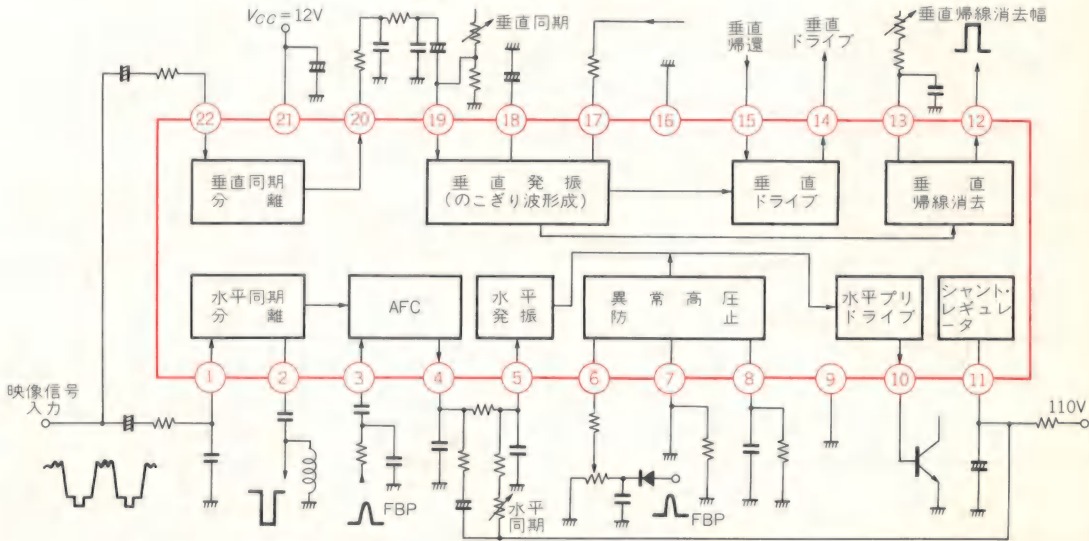


映像コンポーネント信号を3.0V<sub>p-p</sub>で入力する場合, 外部RGB信号の入力は1.0V<sub>p-p</sub>で入力すると, ホワイトのレベルが合うようになっている.

〈図13〉 水平同期型 $4f_{sc}$ VCOのブロック図

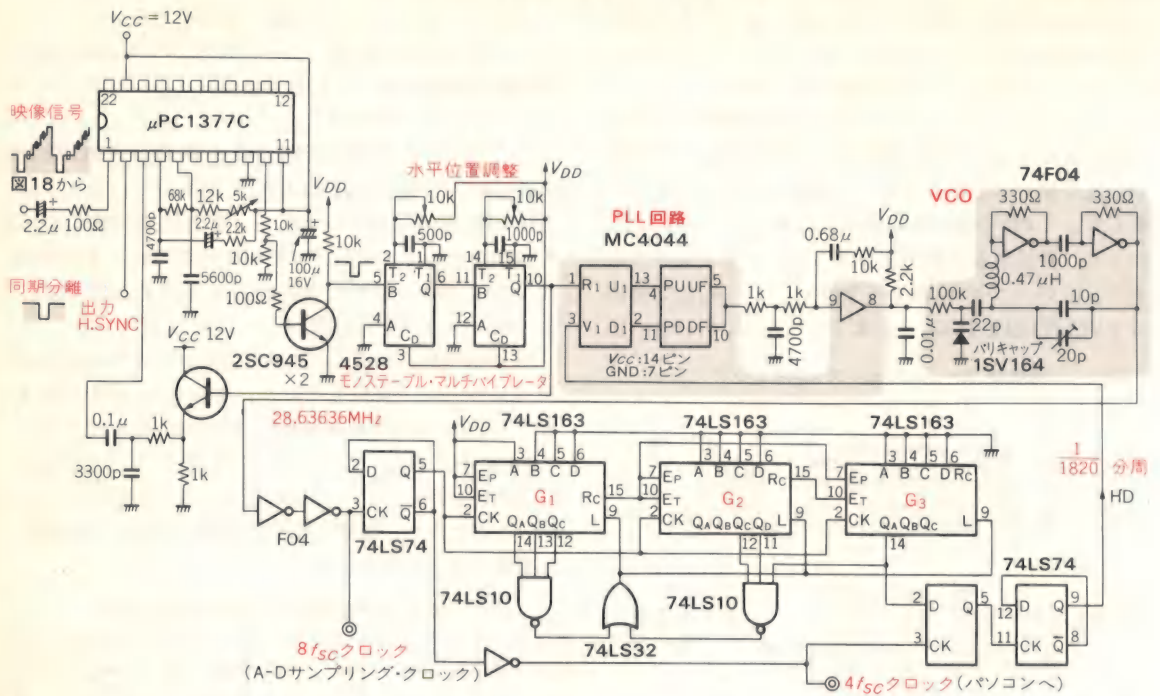


〈図14〉  $\mu\text{PC1377C}$ のブロック図





〈図15〉 水平同期型クロック発生回路



## 21ピン・マルチ・コネクタと8ピン角型コネクタ

画像信号のインターフェースには、EIAJで決められた21ピン・マルチ・コネクタと、パソコンなどで使用する8ピン角型コネクタとがよく使用されま

す。

21ピン・マルチ・コネクタは図AのようにRGB入力、映像、音声入力、テレビ出力(映像・音声)用の21ピン接続端子を備えています。文字多重放送やキャプテ

〈図A〉 21ピン・マルチ・コネクタ

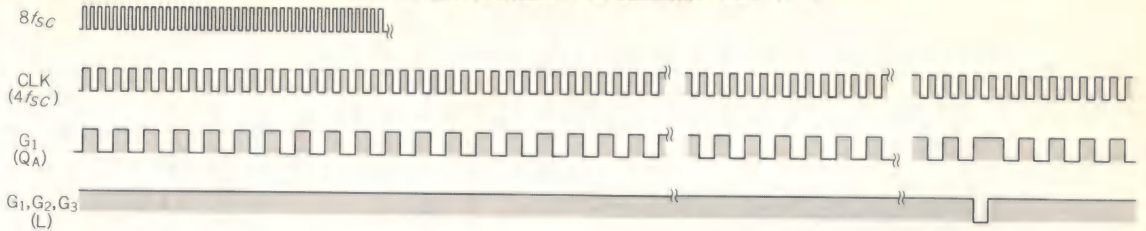
ピンコネクション	端子番号	信号名	端子番号	信号名
	1	音声入力信号(左) 0.4Vrms 入力インピーダンス47kΩ	11	AVコントロール入力信号 低レベル0~0.4Vdc 高レベル3~5Vdc 入力インピーダンス22kΩ
	2	音声出力信号(左) 0.4Vrms 出力インピーダンス10kΩ	12	Yw入力信号 低レベル0~0.4Vdc 高レベル1~3Vdc 入力インピーダンス75Ω
	3	接地	13	接地
	4	接地	14	接地
	5	音声入力信号(右) 0.4Vrms 入力インピーダンス47kΩ	15	赤入力信号 0.7Vp-p 75Ω
	6	音声出力信号(右) 0.4Vrms 出力インピーダンス10kΩ	16	Ys入力信号 低レベル0~0.4Vdc 高レベル1~3Vdc 入力インピーダンス75Ω
	7	接地	17	接地
	8	接地	18	接地
	9	映像/同期入力信号 1Vp-p 75Ω負同期	19	緑入力信号 0.7Vp-p 75Ω
	10	映像/同期出力信号 1Vp-p 75Ω負同期	20	青入力信号 0.7Vp-p 75Ω
			21	ケース接地

〈図B〉  
8ピン角型  
コネクタ

ピン・コネクション	端子番号	信号名
	1	接地
	2	赤映像
	3	緑映像
	4	青映像
	5	接地
	6	接地
	7	水平同期
	8	垂直同期



〈図16〉  $8f_{sc}$  水平同期型クロック発生回路タイムチャート



### ● 水平同期型クロック発生回路

図15が映像信号からPLLによって実現する水平同期型クロックの発生回路です。

この水平同期型クロックのタイムチャートは、図16に示すとおりです。ここで**発振周波数を  $8f_{sc}$  とした理由**は、スーパインポーズだけへの応用ではなく、第9章で紹介するフィールド・メモリのような**RGB信号のA-D変換時のタイミング発生などに応用**ができるようにするためです。

図15の回路の調整方法としては、まず同期負のH. SYNC(水平同期)を入力し、ボリュームの位置はすべてセンタとします。そして**H. SYNCの位相と、MC4044の  $V_1$  入力信号の位相が一致**するようにVCO

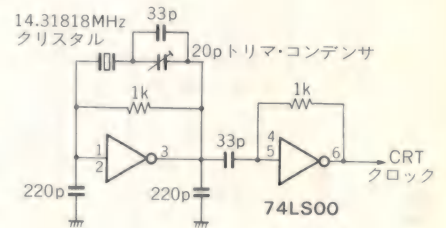
ン・システムに使われるのが現在の主目的ですが、**将来的にはパーソナル・コンピュータ、VTR、ビデオ・ディスク・プレーヤにも多く付いてくるもの**とされます。

このコネクタのうち、RGB入力端子はRGB(75  $\Omega$ )出力端子をもったパーソナル・コンピュータや、文字多重アダプタからの信号を入力します。映像・音声入力端子はVTR、ビデオ・ディスク・プレーヤの映像・音声の信号を入力します。映像・音声出力端子は、テレビ・セット本体の受信チャンネルを常に出し、VTRや情報機器にテレビ番組の信号を出力します。

コントロール入力端子のうち、 $Y_M$ と $Y_S$ は文字多重信号のコントロール信号( $Y_S$ : 内部/外部RGB信号切り替え、 $Y_M$ : シャドウ切り替え)、AVはビデオ入力端子のコントロール信号です。

一方、8ピン角型コネクタは、図BのようにRGB出力端子をもったパーソナル・コンピュータと接続します。このときコンピュータの種類によって信号形式が異なりますので、**データ表示領域のサイズや、データ表示領域の位置がずれて表示する場合**があります。その際には**水平同期ツマミ、垂直同期ツマミ**によって調節する必要があります。

〈図17〉 パソコンのCRTクロック回路例(PC8001)



のトリマ・コンデンサ(20pF)を設定します。これでセッティングは終わりです。

図17に参考のためにパソコンPC8001のCRTクロック回路を示します。このパソコンのクロックの代わりに、図15で作った  $4f_{sc}$  信号を使うと映像信号とパソコン画像との同期がとれることになります。

### ● 実際のスーパインポーズ回路

図18に実際のスーパインポーズの回路を示します。ここでは21ピン・マルチ・コネクタを用い、テレビよりNTSCコンポジット信号を受け、**スーパインポーズ後のRGB信号をテレビに入力する構成**となっています。

コンポジット信号をくし型フィルタでY/C分離した後、 $\mu PC1352C$ でコンポジット信号→RGBコンポーネント信号への変換を行い、 $\mu PC1397C$ でスーパインポーズするようになっています。なお、この時のスーパインポーズのタイミングは先の図15の水平同期型クロック発生回路を使用します。

スーパインポーズを使用しなければ、図15の水平同期型クロック発生回路は不要で、図18の回路のままでコンポジット→RGB信号コンバータとして使用することができます。

このコンポジット→RGB信号コンバータを使えば、RGBディスプレイによる鮮明なテレビ画像を楽しむこともできますし、**画像処理システムでのデジタル化前処理としての用途**へ利用できます。〈松井俊也〉

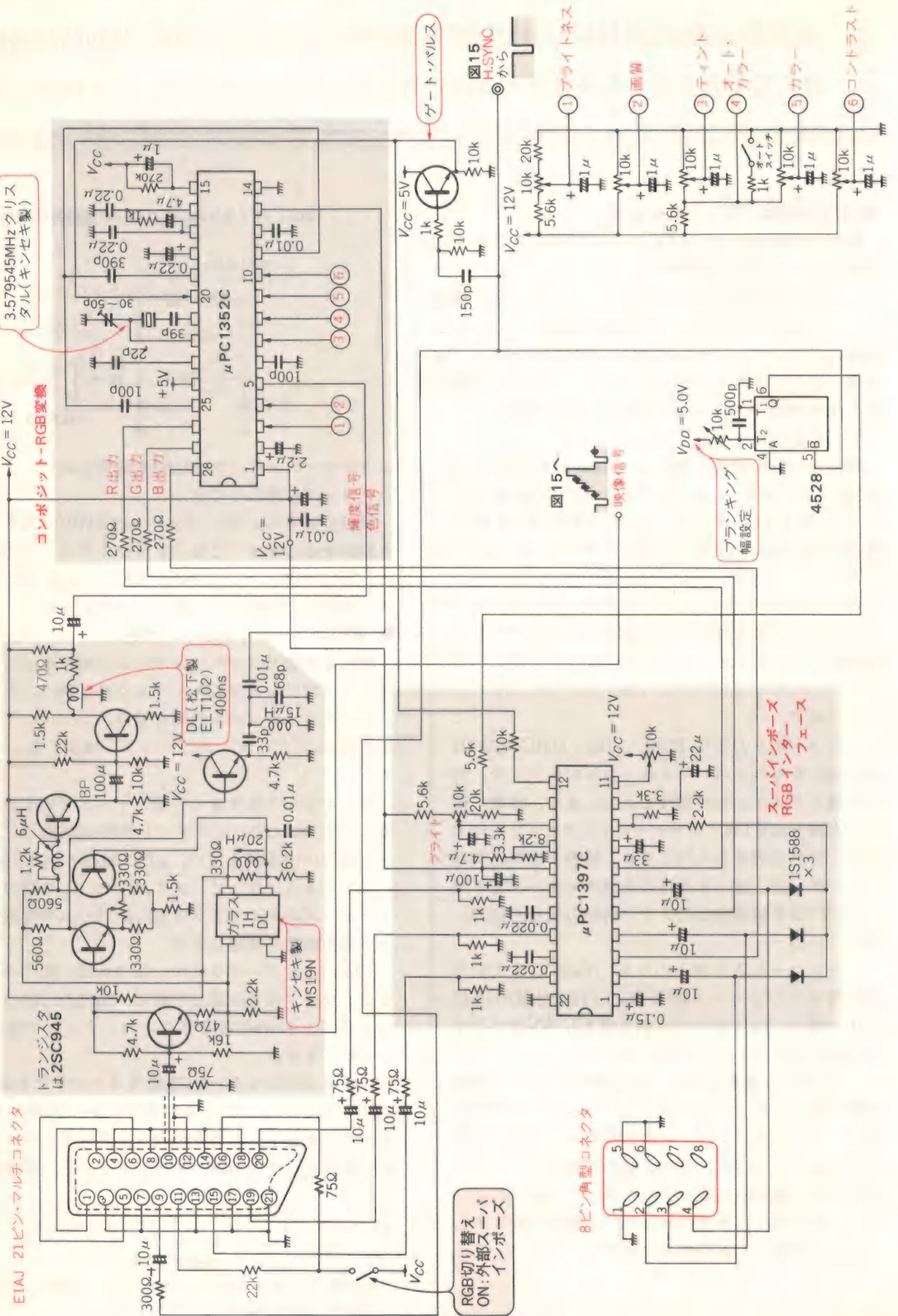
### ● 参考文献

- (1) 赤羽根 仁：映像信号のインターフェース技術，トランジスタ技術，1986年6月号。



〈図18〉 コンポジット-RGBコンバータ (スーパーインポーズ付き) 回路

くし型フィルタ (Y/C分離回路)





## カラー・スーパインポーズの設計・製作



千葉雅彦  
峯岸英雄

スーパインポーズした文字に任意の着色ができると画像の編集効果が高まります。ここではコンポジット信号の中にカラーでスーパする例を紹介します。巻頭のカラー口絵も参照してください。

最近のテレビやVTRでは、画面の中にチャンネル、音量、時刻などの表示ができるものが増えてきました。これが、文字のスーパインポーズと呼ばれるものです。

トランジスタ技術本誌でも、すでに何度かスーパインポーズの原理、設計・製作記事が紹介されていますが、ここではスーパインポーズされた文字に好みの色を付ける、すなわちカラー・スーパインポーズの設計を行います。

### スーパインポーズのしくみ

#### ● スーパインポーズの原理

ここでは、文字のスーパインポーズの原理を説明します。図1が基本原理を示したものです。

最近のテレビに付属しているビデオ出力端子、またはVTRのビデオ出力端子には、コンポジット・ビデオ信号と呼ばれる、複合映像信号が出力されています。これは、図2で示すように、画像情報を伝送し、かつブラウン管上に表示するための情報を含んでいます。この中の水平同期信号と垂直同期信号が、テレビ画面上のどの部分の画像情報かを知らせるための信号です。

スーパインポーズでは、映像信号とキャラクタ・ジ

ェネレータとの同期をとるために、コンポジット・ビデオ信号から、この同期信号を抜き取ります。そして、これをキャラクタ・ジェネレータに加えることにより、映像信号と同期のとれた文字信号を発生させ、図1のビデオ・スイッチを文字信号のある所だけDC成分(輝度レベル)に切り替えるようにすれば、文字、記号などをテレビ画面上に表示できるようになります。この時の各信号のタイミング関係を図3に示します。

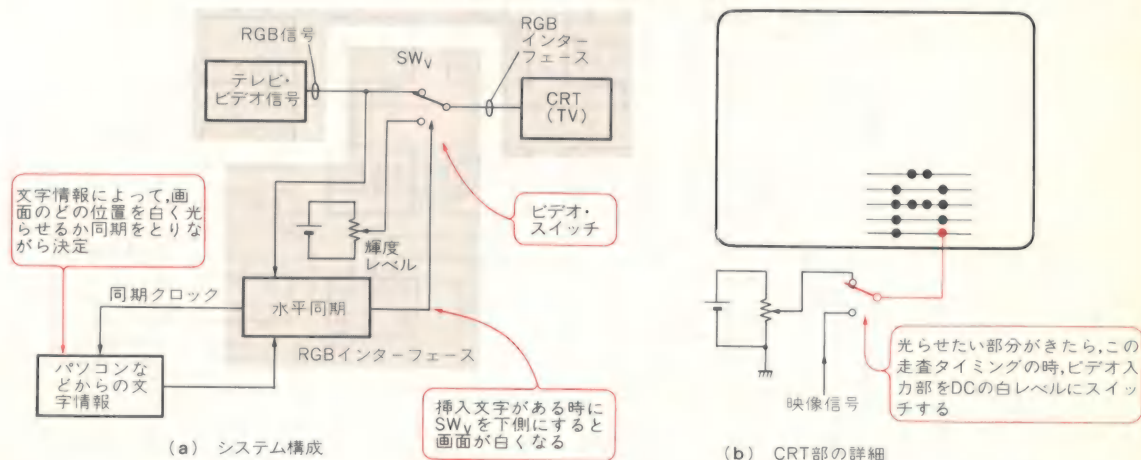
キャラクタ・ジェネレータの内部ブロックの一例を図4に示します。このICはクロック発生回路を内蔵しており、同期分離されたコンポジット同期信号の中の水平同期信号の立ち上がり、または立ち下がりエッジでトリガがかかり、クロックの発振を開始します。このクロックの周期が文字の水平、垂直方向の位置、及び文字寸法の基準となります。

#### ● カラー・スーパインポーズを実現するには

図1の原理図にある方法だと、スーパインポーズした文字、記号の色は、DC電圧(輝度レベル)の可変しかできません。つまり、白、黒、灰色といった色しか付けることができません。では、スーパインポーズした文字に赤、青といった色を付けるにはどうしたらよいのでしょうか。

これは、図5のような方法で簡単に行うことができ

〈図1〉文字スーパインポーズの原理

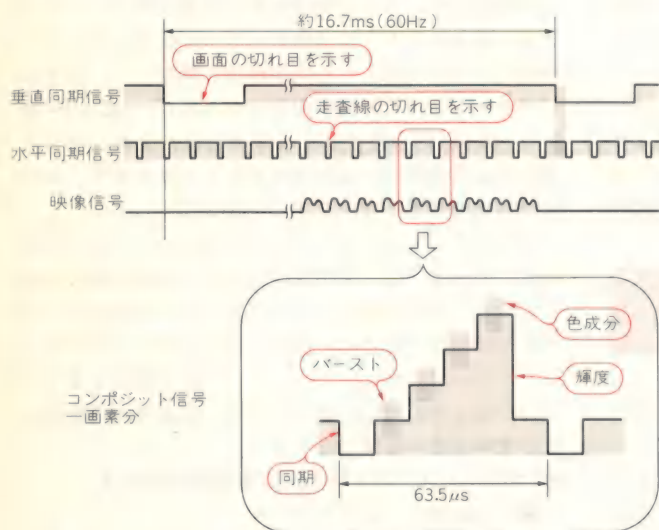




ます。つまり、ビデオ・スイッチで切り替えて加えるDC電圧の代わりに、ある**任意のDCオフセットをもたせた、3.58MHzの正弦波**(図2の色信号成分に相当する)を加えてやればよいわけです。

ただし、ここで注意しなければならないことは、コンポジット・ビデオ信号中の**カラー・バースト信号と、加える3.58MHzの信号の位相がロック**していなければならないということです。バースト信号に対して、3.58MHz成分の位相が同期せずに動いてしまうと、スーパインポーズした文字の色がクルクルと変化してしまいます。

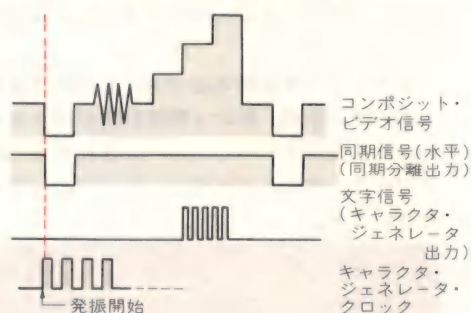
〈図2〉コンポジット・ビデオ信号



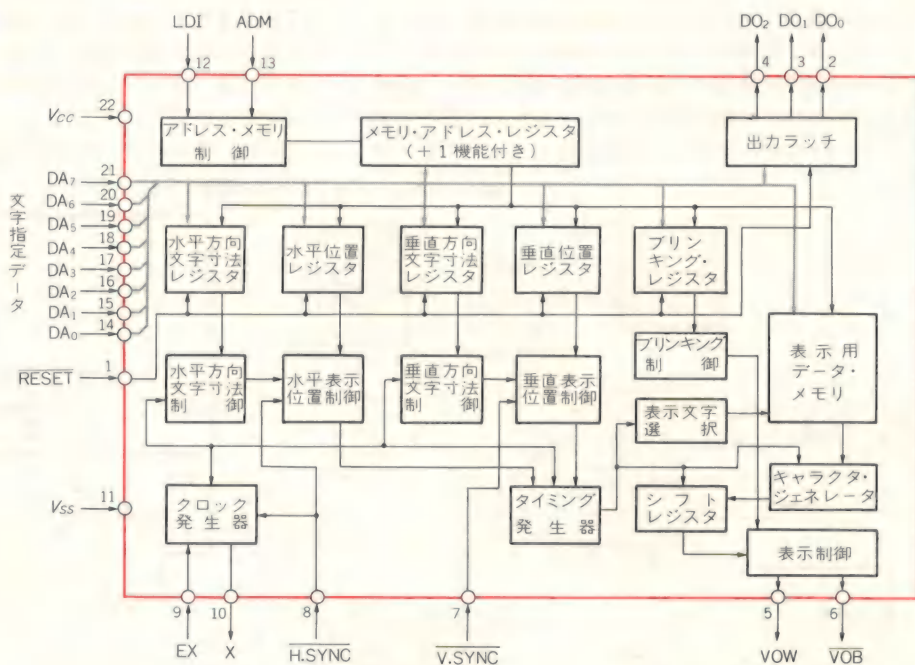
## スーパインポーズ設計の基本

スーパインポーズの基本システム・ブロックは図6に示すように、大きく**ビデオ・スイッチ**、**キャラクタ・ジェネレータ**、および**同期分離回路**に分けられます。このうちキャラクタ・ジェネレータについては、ほかの記事でも取り上げられる機会が多いので、ここではビデオ・スイッチと同期分離回路の基本動作と設計上のポイントについて説明したいと思います。

〈図3〉コンポジット・ビデオ信号とキャラクタ・ジェネレータ出力



〈図4〉<sup>(1)</sup>キャラクタ・ジェネレータの内部構成例(MB88303, 富士通)





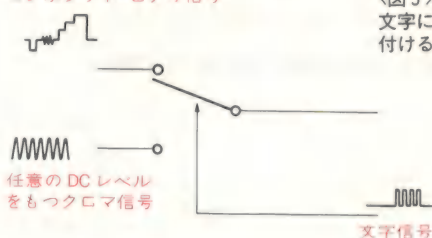
## ● ビデオ・スイッチのしくみ

ビデオ・スイッチの基本回路を図7に示します。二つの信号をそれぞれトランジスタ $Q_3$ 、 $Q_6$ のベースに入力し、 $Q_2$ のベースにコントロール信号を入力します。そして、 $Q_1$ と $Q_2$ のどちらかに電流が流れるようにすることによって、 $Q_3$ と $Q_4$ または $Q_5$ と $Q_6$ のそれぞれの差動ペアに流す電流を切り替えて、出力に取り出す信号をスイッチします。

スーパインポーズで用いる場合には、 $v_1$ にビデオ信号を入力し、 $v_2$ に一定のDC電圧を与えます。そして、コントロール端子にキャラクタ・ジェネレータからの文字信号を加えるようにします。こうすれば、文字信号が“H”の時に、スイッチ出力には $v_2$ からのDC電圧が出力され、“L”の時は $v_1$ のビデオ信号が出力されることになります。

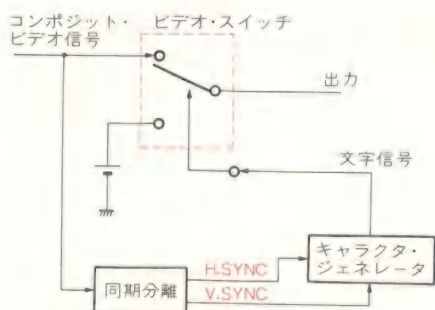
この場合に注意しなければならない点は、ビデオ・スイッチに入力されるビデオ信号が、あるDC電圧でクランプされていないと、このクラ

コンポジット・ビデオ信号

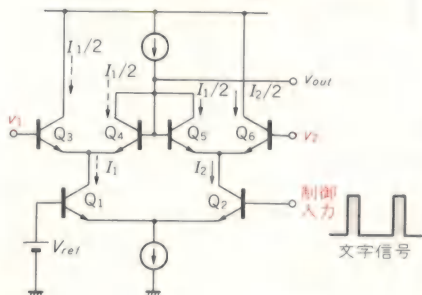


〈図5〉  
文字に色を  
付ける方法

〈図6〉 スーパインポーズの  
基本ブロック



〈図7〉 ビデオ・スイッチ基本回路



ンプ電圧と文字の輝度レベルを与えるDC電圧の関係です。これを図8に示します。

いま、コンポジット・ビデオ信号が水平同期部分の底でクランプ(シンクチップ・クランプ)されているとすると、文字信号の輝度レベルとして与えるDC電圧は、ビデオ信号のペダスタル・レベルに相当する電圧以上でなければなりません。

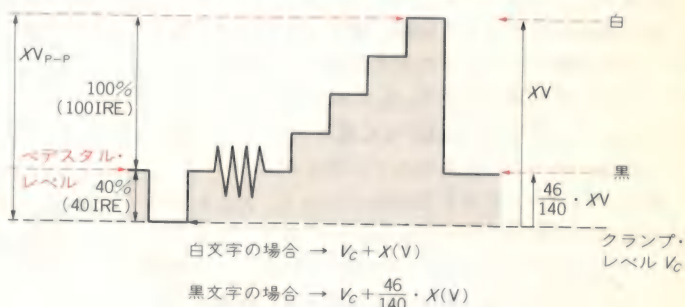
## ● 同期分離の方法

キャラクタ・ジェネレータから、画像と同期の合った文字信号を取り出すためには、トリガとなる信号が必要です。これが水平同期信号と垂直同期信号です。そして、コンポジット・ビデオ信号から水平、垂直同期信号を取り出すのが同期分離回路です。

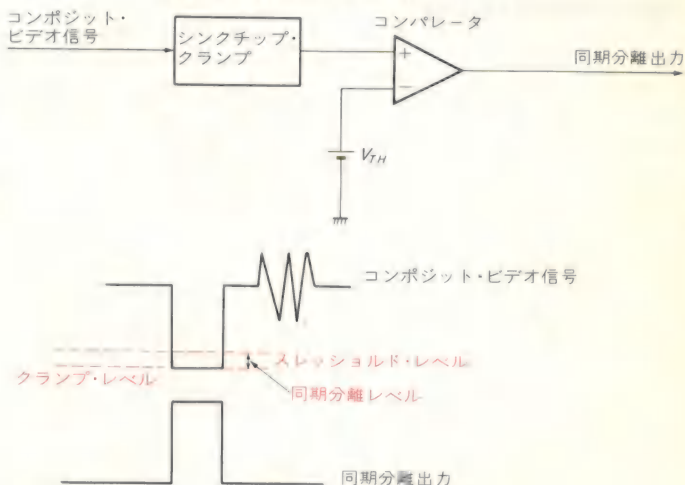
同期分離回路の動作原理を図9に示します。同期分離回路は、シンクチップ・クランプ回路とコンパレータから構成されています。

コンポジット・ビデオ信号は、シンクチップ・クランプ回路で同期成分の底の部分があるDC電圧で固定(クランプ)され、コンパレータに加えられます。そして、コンパレータでクランプ・レベルから適当な電圧の所にスレッシュホールド・レベルを設定しておけば、同期成分だけ取り出せるわけです。クランプ・レベル

〈図8〉 クランプ・レベルと文字の輝度レベル



〈図9〉 同期分離回路の動作原理





とスレッショルド・レベルとの差電圧を同期分離レベルと呼んでいます。

同期分離回路を用いるうえで注意しなければならない点は、この同期分離レベルです。同期分離回路に加えられるコンポジット・ビデオ信号の同期成分の振幅の大小にかかわらず、シンクチップから一定のレベルで同期分離しますので、同期分離回路に加えられるコンポジット・ビデオ信号の振幅によって、同期分離レベルを設計する必要があります。

一般的には、コンポジット同期成分の振幅の20%から25%程度に同期分離レベルを設計しておけば問題ないでしょう。コンポジット・ビデオ信号が $2V_{P-P}$ の場合、同期信号の振幅は $0.57V_{P-P}$ となっていますので、 $110mV_{P-P}$ から $140mV_{P-P}$ となります。

## スーパインポーズの回路設計

ここでは、スーパインポーズする文字に色を付けることができる、色発生機能をもつNJM2214を用いたスーパインポーズの構成方法について説明します。

### ● NJM2214の基本動作

NJM2214は図10に示すように、4個のビデオ・スイッチと8色発生回路、輝度信号整形回路より構成されています。

色の発生はマトリクス回路、電子ボリューム、Yレベル加算回路からなる8色発生回路で行います。

マトリクス回路は $S_1$ と $S_2$ 、 $S_3$ に加えられる3ビットのデジタル信号を、任意のDC電圧 $V_1$ 、 $V_2$ に変換し、二つの電子ボリュームにそれぞれ加えます。

電子ボリュームには $3.58MHz$ の $f_{sc1}$ と、 $f_{sc1}$ より $90^\circ$ 位相の遅れた $f_{sc2}$ がそれぞれ加えられます。そして、マトリクス回路からの $V_1$ と $V_2$ に応じたゲインで増幅されたあと合成され、Yレベル加算回路で輝度レベル

を与えられてクロマ信号となります。

### ● 色の3属性を実現するしくみ

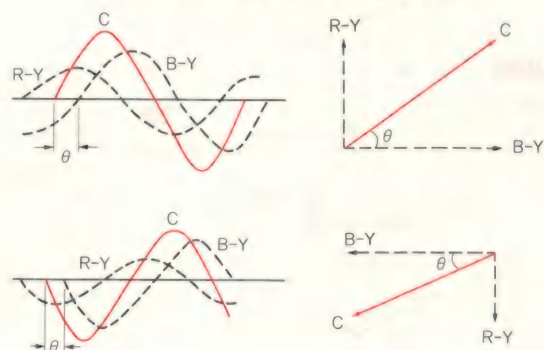
ここで、カラー信号について簡単に述べておきます。色には色の3属性と呼ばれる次の三つの性質があります。

- (1) 色相：赤、青といった色あい
- (2) 飽和度：色の濃さ、あざやかさ
- (3) 明るさ：輝度

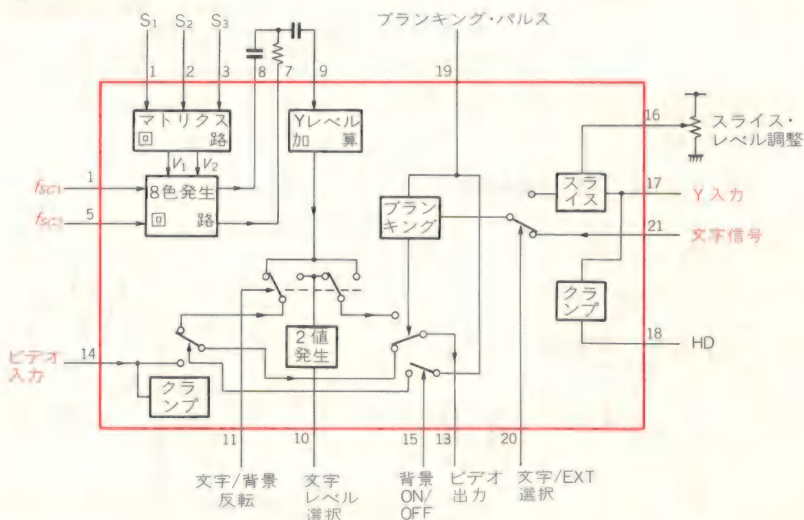
ビデオ信号も色信号を伝送するためには、この三つの情報をもっていなければなりません。そして、コンポジット・ビデオ信号の中では、クロマ信号の振幅が飽和度の情報を伝送するために使われています。また、バースト信号との位相差が色相の情報を、輝度信号が明るさの情報を伝送するために使われています。

それでは、クロマ信号を作るにはどうしたらいいのでしょうか。現在のカラー・テレビジョン方式では、クロマ信号を作るために互いに $90^\circ$ 位相差をもった $3.58MHz$ の搬送波を、色差信号 $E_R - E_Y$ 、 $E_B - E_Y$ でそれぞれ振幅変調して合成する方法を取っています。この原理を図11に示します。図11でわかるように、ク

〈図11〉 クロマ信号と二つの搬送波の関係



〈図10〉<sup>(3)</sup> NJM2214ブロック図





ロマ信号の振幅および位相差を、色差信号の大きさだけで変えることができます。

この色差信号に相当するものが、NJM2214では $V_1$ と $V_2$ です。そして $V_1$ と $V_2$ に応じて、それぞれの搬送波の振幅を変える働きをしているのが二つの電子ボリュームです。

マトリクス入力と色との対応を表1に示します。

● 4個のビデオ・スイッチの働き

NJM2214では4個のビデオ・スイッチが図12のように接続されています。例えば、 $SW_1$ がB、 $SW_3$ と $SW_4$ がともにBの場合、 $SW_1$ のCには8色発生回路からのカラー信号が出力されます。そして、 $SW_2$ のBには $V_w$ と $V_b$ のDC電圧が加えられています。

ここで、文字信号が入力されると、 $SW_2$ のCには色の付いた背景に白( $V_w$ )または黒( $V_b$ )の文字がスーパーインポーズされます。

次に $SW_1$ をA、 $SW_3$ と $SW_4$ をAにすると、 $SW_2$ のBにはカラー信号が来ていますので、通常の映像に色の付いた文字がスーパーインポーズされることになります。

このように、スイッチに通す信号の組み合わせで、いろいろな形のスーパーインポーズを行うことができま

す。制御入力端子の状態と機能の対応を表2に示します。

● 輝度信号の整形機能

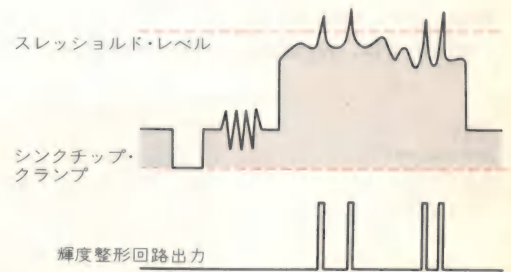
最後に、輝度信号の整形機能について説明します。これは、ビデオ信号中の輝度の高い部分だけを取り出してパルス出力として、文字信号と同じように扱うことができる機能です。

この原理は同期分離と同じです。コンポジット・ビデオ信号をあるDC電圧でクランプし、スレッショルド・レベル以上の輝度レベル部分でのみ、コンパレータ出力がアクティブになるようになっていきます。この動作原理図を図13に示します。輝度の高い部分のみを文字信号と同様に扱うことができるわけですから、この部分に色を付けたりすることができます。すなわち、簡単なペイント効果を通常の再生映像に与えることができます。

● 実際のシステム構成

NJM2214を用いたカラー・スーパーインポーズのシス

〈図13〉 輝度整形回路の動作原理



〈表1〉 8色出力とマトリクス入力の関係

色	マトリクス1	マトリクス2	マトリクス3
白	L	L	L
黄	H	L	L
シアン	L	H	L
緑	H	H	L
マゼンタ	L	L	H
赤	H	L	H
青	L	H	H
黒	H	H	H

L=0V, H=5V

〈表2〉 制御入力と機能

背景ON/OFF	文字 2 値選択	文字背景反転	EXT/文字選択	内 容
L	L/H	L	L	ビデオ・スルー出力に文字のスーパーインポーズ (白/黒)
H	L/H	L	L	背景 (8 色) に文字 (白/黒) のスーパーインポーズ
H	L/H	H	L	背景 (白/黒) に色のついた文字
L	L	H	L	ビデオ・スルーに色のついた文字
L	L/H	L	H	輝度の高い所が白または黒
H	L/H	L	H	輝度の高い所以外に色がつく
H	L/H	H	H	輝度の高い所に色がつき、それ以外は白または黒
L	H	H	H	輝度の高い所だけ色がつき、他はビデオ・スルー



テム・ブロック図を図14に示します。

NJM2214を用いてカラー・スーパーインポーズの動作を行うためには、同期信号のほかに、入力されるコンポジット・ビデオ信号のカラー・バースト信号と同期した3.58MHzのサブキャリア $f_{sc1}$ 、 $f_{sc2}$ と、ブランキング・パルスが必要になります。また、輝度信号の整形を行うためにHD(水平・ドライブ)パルスも必要になります。

# ● スーパーインポーズのための信号を作る

まず、スーパーインポーズに必要な信号を作り出すシグナル・ジェネレータについて説明します。今回は、 $f_{sc1}$ と $f_{sc2}$ を作るために、VTRカラー信号処理用ICの $\mu$ PC1536を用います。また、同期分離ICとして、NJM2217を用います。

$\mu$ PC1536のブロック図を図15に、NJM2217のブロック図を図16に示します。

シグナル・ジェネレータの回路は図17です。また、図17のポイントとなる波形を写真1～写真15に示します。VIDEO-INに入力されたコンポジット・ビデオ信号は、まずY/C分離回路に加えられます。そしてY/C分離回路で、輝度信号とクロマ信号に分けられます。

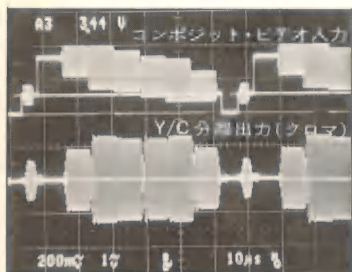
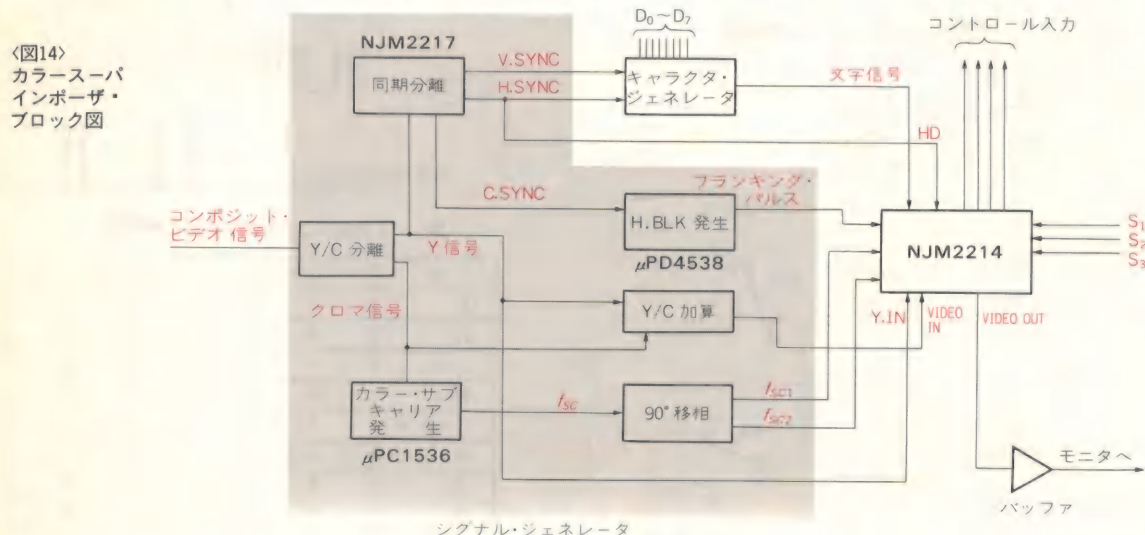
輝度信号は、NJM2217の同期分離入力端子に加え

られ同期分離されます。同期分離されたコンポジット同期信号(C. SYNC)は、NJM2217内部の垂直同期再生回路に積分回路を通して加えられます。また、 $\mu$ PC1536およびH. BLKパルス発生回路に送られます。H. SYNC、V. SYNC出力はキャラクタ・ジェネレータに送られます。そしてH. SYNCはNJM2214のHDパルスとしても使います。

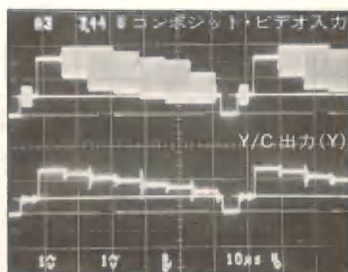
NJM2217のAFC回路を入力信号に対して周波数ロックさせるために、 $SW_2$ をONして $VR_1$ を調整して、H. SYNC出力の周波数を15.734kHzに合わせます。

H. BLKパルス発生回路では、ワンショット・マルチバイブレータを4個用いて、NJM2214のビデオ・スイッチをコントロールするために必要なブランキング・パルスを作ります。NJM2214のビデオ・スイッチの動作とブランキング・パルスの関係を図18に示します(写真8)。

$\mu$ PC1536では、Y/C分離回路からのクロマ信号中のカラー・バースト信号と、位相がロックした3.58MHzのサブキャリア信号 $f_{sc}$ を作ります。そして、 $f_{sc}$ を90°移相回路に加えて、 $f_{sc}$ と同相の $f_{sc1}$ と $f_{sc2}$ から90°位相の遅れた $f_{sc2}$ を作り、NJM2214に加えます。この $f_{sc1}$ と $f_{sc2}$ が、NJM2214がIC内部で作るカラー信号の



〈写真1〉



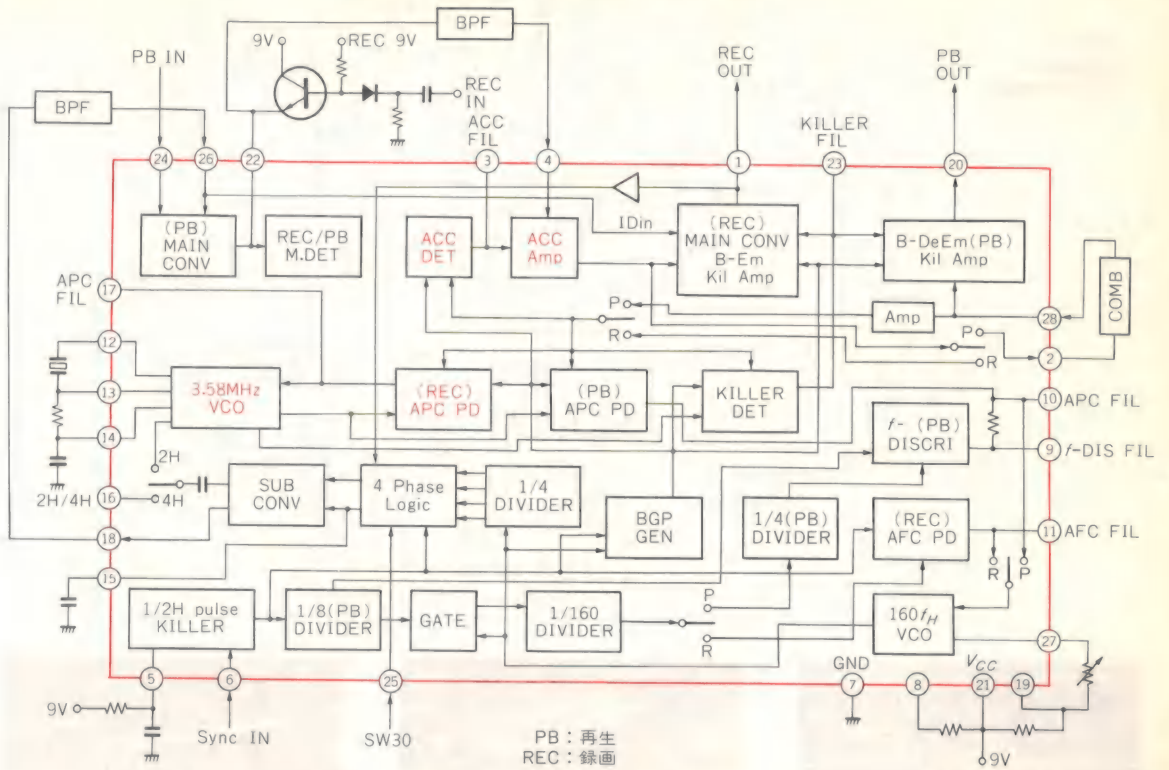
〈写真2〉



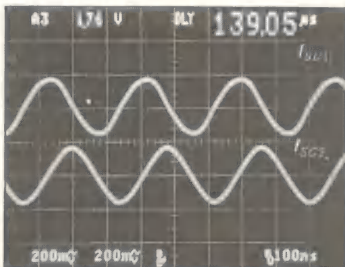
〈写真3〉



〈図15〉<sup>(12)</sup> カラー信号処理IC  $\mu$ PC1536Cの構成



- |                                   |    |                                  |    |
|-----------------------------------|----|----------------------------------|----|
| REC Color 出力(REC時)                | 1  | (PB)Burst Det-emphasis           | 28 |
| ID入力(PB時)                         | 2  | Killer Amp入力(1HDLより)             | 27 |
| Buffer Amp出力(1HDL Filterへ)        | 3  | 160f <sub>H</sub> VCO制御電流入力      | 26 |
| (PBのみ)                            | 4  | Main Conv入力(周波数変換用キャリア)          | 25 |
| ACC Det Filter 端子                 | 5  | 30Hz Pulse信号入力                   | 24 |
| ACC Amp入力                         | 6  | Main Conv入力(再生629kHz信号)(PBのみ)    | 23 |
| 1/2H Pulse Killer回路のモノマルチ時定数端子    | 7  | Killer Det Filter 端子             | 22 |
| 1/2H Pulse Killer回路入力(Sync Pulse) | 8  | (REC)REC Color信号入力(3.58MHz BPFへ) | 21 |
| GND                               | 9  | (PB)Main Conv出力                  | 20 |
| I <sup>2</sup> Lインジェクタ端子          | 10 | V <sub>CC</sub>                  | 19 |
| Freq Discr Filter 端子              | 11 | PB Color出力(PBのみ)                 | 18 |
| PB時 APC PD Filter 端子              | 12 | ツェナ電圧端子                          | 17 |
| REC時 AFC PD Filter 端子             | 13 | Sub Conv出力                       | 16 |
| 3.58MHz VCO出力                     | 14 | REC時APC PD Filter端子              | 15 |
| 3.58MHz VCO入力                     | 15 | 3.58MHz VCO出力                    | 14 |
| 3.58MHz VCO入力                     | 16 | 4相40f <sub>H</sub> 信号バイパス        | 15 |



〈写真4〉



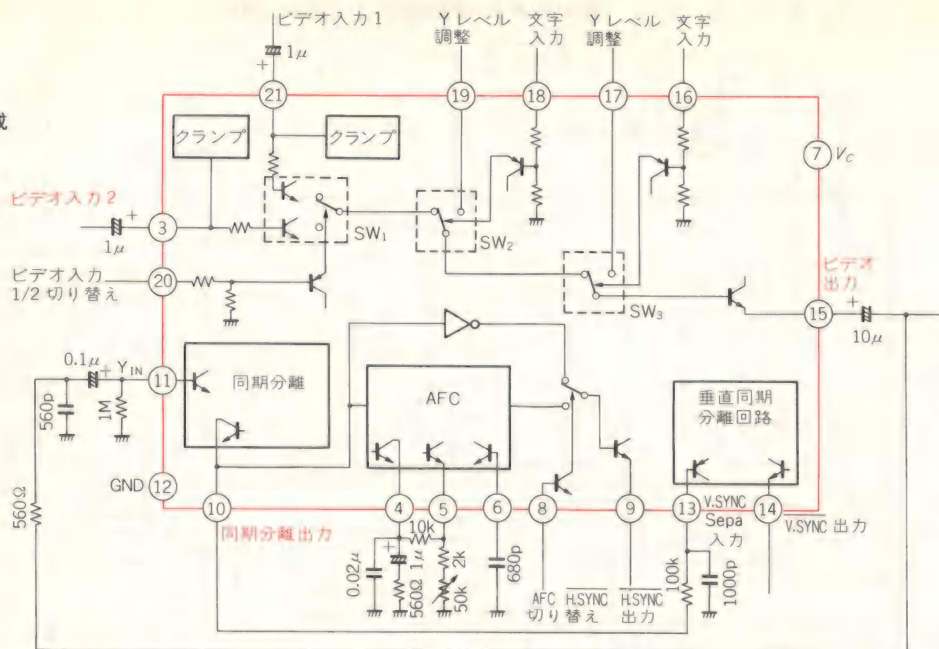
〈写真5〉



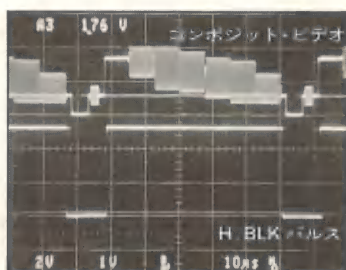
〈写真6〉



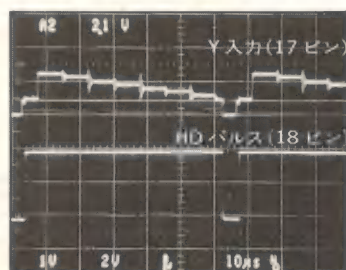
〈図16〉<sup>(4)</sup>  
同期分離IC  
NJM2217の構成



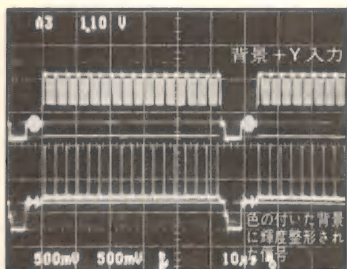
〈写真7〉



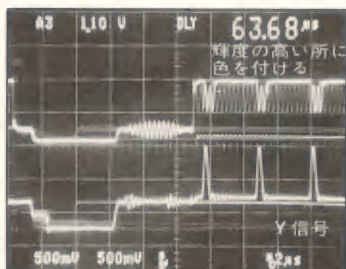
〈写真8〉



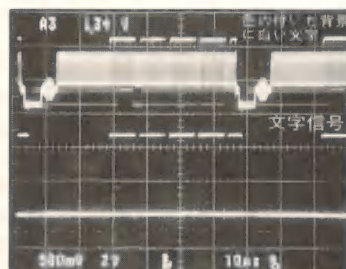
〈写真9〉



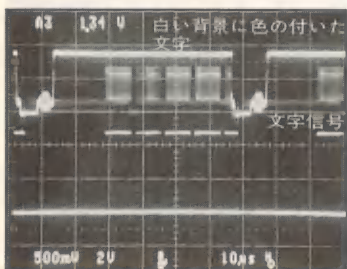
〈写真10〉



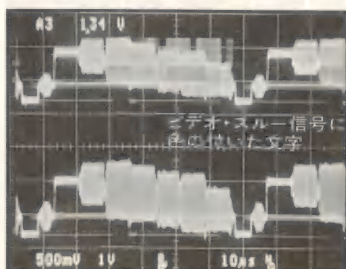
〈写真11〉



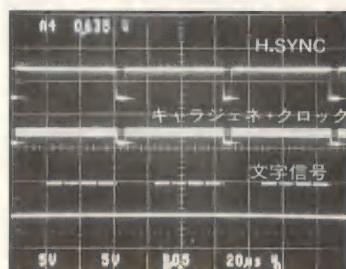
〈写真12〉



〈写真13〉



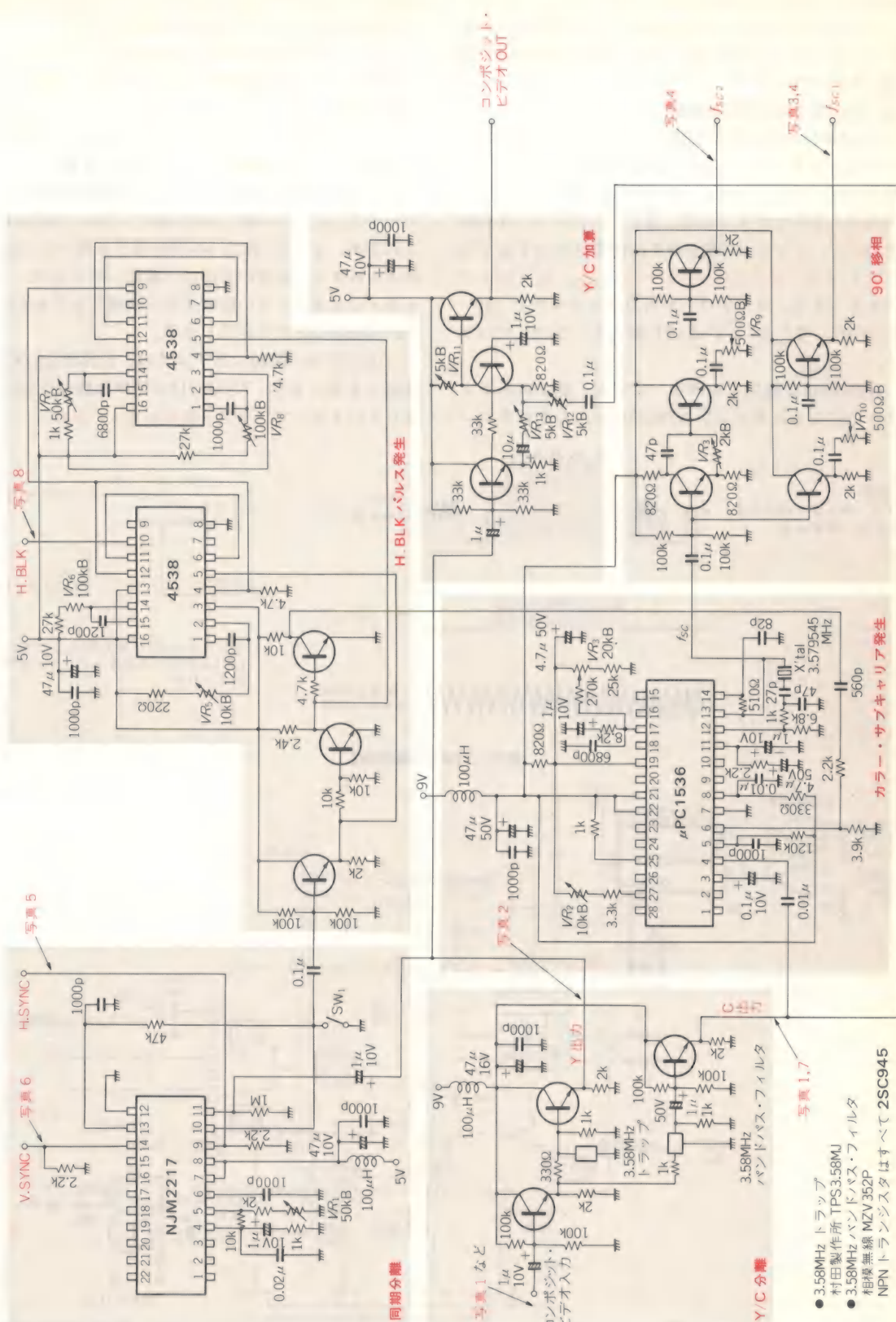
〈写真14〉



〈写真15〉



＜図17＞ シグナル・ジェネレータ回路



- 3.58MHz トラップ  
村田製作所 TPS3.58MJ
  - 3.58MHz バンドパス・フィルタ  
相模無線 MZV 352P
- NPN トランジスタはすべて 2SC945



サブキャリアとなります。

このとき、 $VR_9$ と $VR_{10}$ で $f_{SC1}$ と $f_{SC2}$ の振幅が300 mV<sub>P-P</sub>になるように調整します。また、90°の位相調整は $VR_3$ で行います。

### ● NJM2214の周辺回路設計

NJM2214の周辺回路を図19に示します。色をコントロールする3ビットのデジタル信号は、マイコンなどでコントロールすると便利です。ここでは、システムが大きくなりますので、3ビットのカウンタ回路を用いて、プッシュSWを押す回数を0から7までカウントして、出力をNJM2214の $S_1$ 、 $S_2$ 、 $S_3$ に加えています。また、コントロール入力はスイッチで“H” (= 5V)、“L” (= 0V)を切り替えるようにしています。

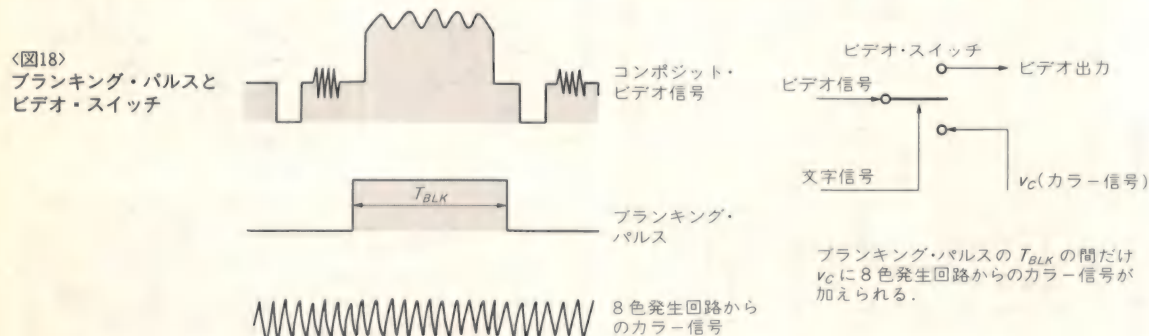
NJM2214で発生するカラー信号が、表1にあるような色になるためには、NJM2214に入力されるコン

ポジティブ・ビデオ信号のカラー・バースト信号と、 $f_{SC1}$ の位相が同相になっていることが必要です。

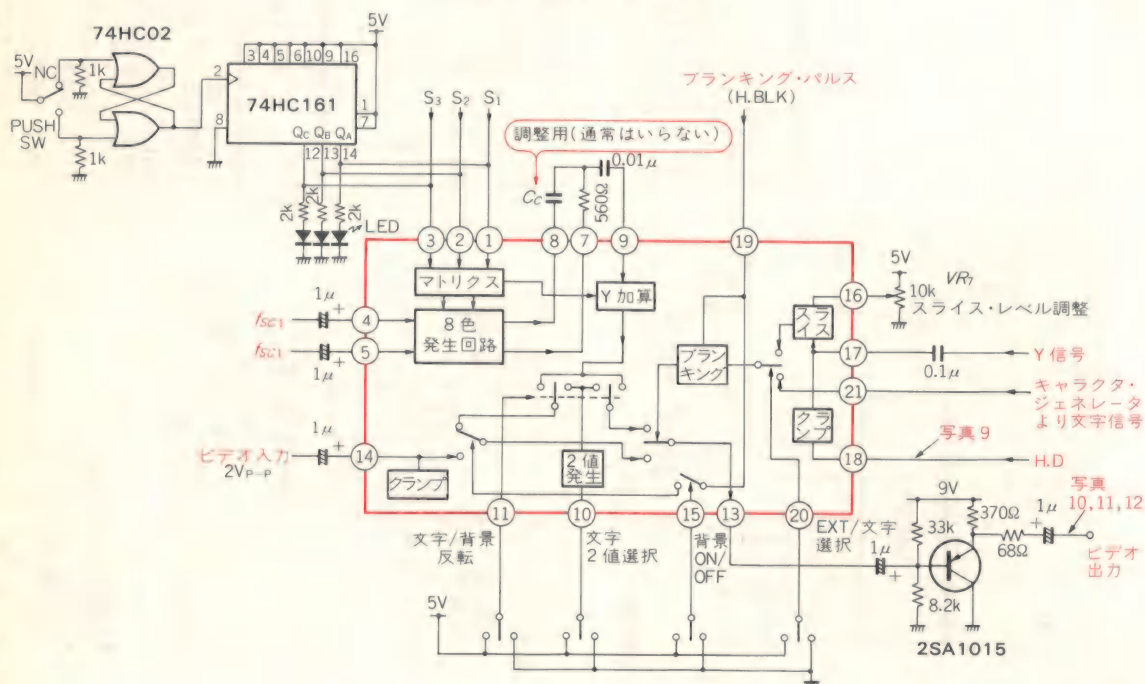
しかし、Y/C分離回路の3.58MHzバンドパス・フィルタで、フィルタの群遅延によるバースト信号の位相遅れが発生します。したがって、シグナル・ジェネレータに入力されるバースト信号と $f_{SC1}$ とは、周波数はロックしていても同相にはなりません(写真7)。

この場合の調整方法としては、90°移相回路の $f_{SC}$ についても、 $f_{SC2}$ と同様の移相回路を入れる方法が考えられます。またNJM2214の調整端子を用いる方法も考えられます。前者の方法が一番良いのですが、シグナル・ジェネレータの回路が多少複雑になりますので、今回は後者の方法を紹介します。

この方法は、図20に示すように、NJM2214の⑦、⑧端子を用います。NJM2214は簡単な移相回路を内蔵していますので、カラー信号を作ってから、その位



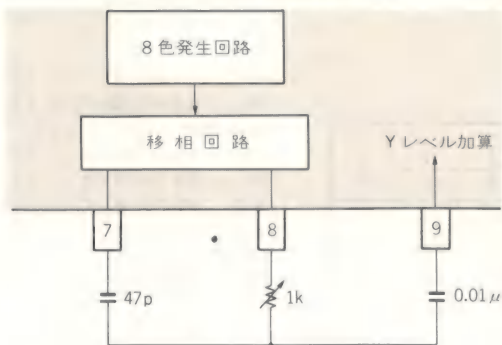
〈図19〉 NJM2214周辺回路



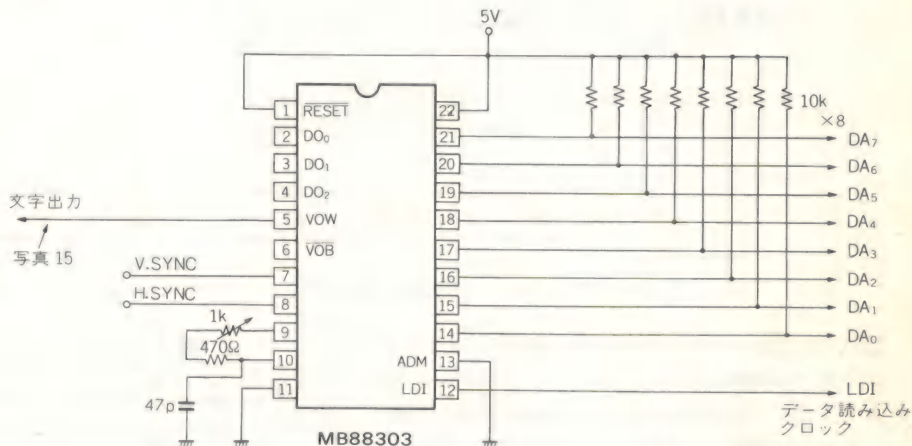


相を動かすわけです。この方法は、ベクトル・スコープなどの測定器がある場合は、それを用いて各色の位

〈図20〉 カラー信号位相調整方法



〈図21〉 文字信号発生回路



相が合うように調整すれば良いのですが、測定器がない場合は、NJM2214のS<sub>1</sub>とS<sub>2</sub>、S<sub>3</sub>を黄色出力状態にして、モニタ上で黄色に見えるように⑦、⑧端子の定数を調整してください。

#### ● キャラクタ・ジェネレータについて

スーパインポーズする文字信号を作るには、キャラクタ・ジェネレータを用いるのが一般的な方法です。ここでは、MB88303を用いた文字信号発生回路を説明します。MB88303の内部ブロックは図4に示してあります。

文字信号発生回路を図21に示します。MB88303にはROM型のキャラクタ・ジェネレータが内蔵されています。このキャラクタ・コード一覧を表3に、また表示メモリの構成を図22に示します。

## ▶▶▶ CORE BOOKS

CQ出版社

マイコン活用のためのハードウェア技術入門  
実用インターフェース設計法

畔津 明仁 著  
A 5 判・212頁  
定価 1,440円



安定に、正確に、効率よくまわす技術  
DCモータの制御回路設計

谷腰 欣司 著  
A 5 判・200頁  
定価 1,550円



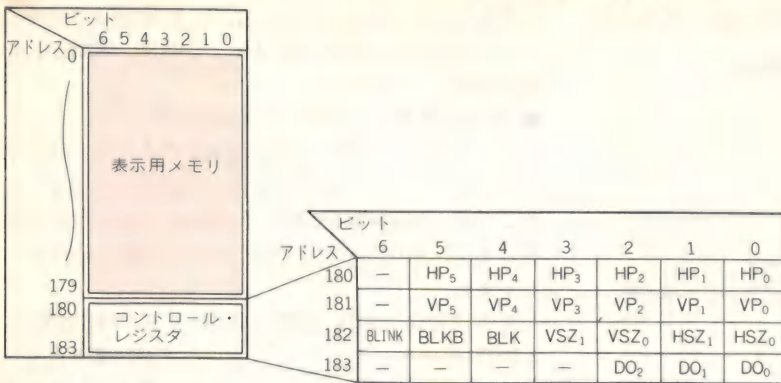
実験で学ぶTTL、C-MOSの応用テクニック  
デジタルIC回路の設計

湯山 俊夫 著  
A 5 判・256頁  
定価 1,650円





〈図22〉メモリ構成



- ・HP<sub>5-0</sub> : 水平表示位置レジスタ 6ビット
- ・VP<sub>5-0</sub> : 垂直表示位置レジスタ 6ビット
- ・表示・文字コントロールレジスタ
  - HSZ<sub>1-0</sub> : 水平方向文字寸法 2ビット
  - VSZ<sub>1-0</sub> : 垂直方向文字寸法 2ビット
  - BLK : 表示ブランキング 1ビット
  - BLKB : 背景ブランキング 1ビット
  - BLINK : ブリンキング 1ビット
  - DO<sub>2-0</sub> : 汎用出力レジスタ 3ビット

〈図23〉データの書き込み

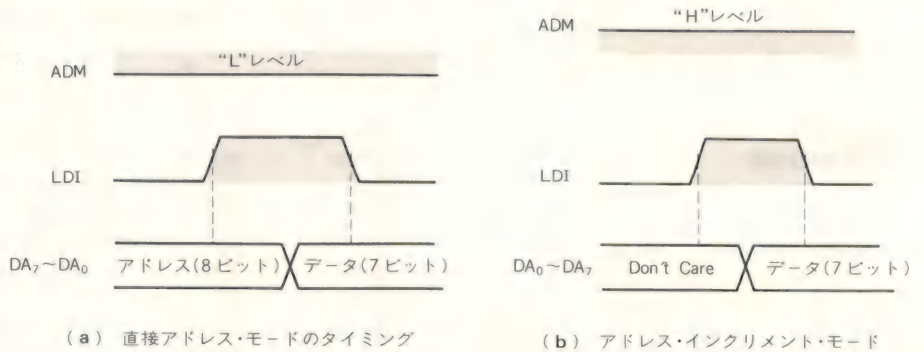


図21で、DA<sub>0</sub>からDA<sub>7</sub>にデータを送り、表示用メモリにデータを書き込むことによって、文字信号を発生します。データ書き込みのモードを図23に示します。

これまでに説明してきたブロックを組み合わせることによって、カラー・スーパーインポーズの基本システムを構成することができます。また、NJM2214のもつカラー・ペイント効果を用いて、テープ編集を行う場合の特殊効果を加えることもできます。

#### ● 製作上の注意点

製作上の注意点は、アナログ系とデジタル系のまわり込みに注意することです。グラウンドはアナログ、デジタル系で分離します。また電源ラインは、アナログ系の各IC、ブロックごとにLCによるデカップリング・フィルタを入れることが大切です。

#### ● 参考文献 ●

- (1) MB88303データ・シート、富士通㈱。
- (2) μPC1536Cデータ・シート、日本電気㈱。
- (3) NJM2214データ・シート、新日本無線㈱。
- (4) NJM2217データ・シート、新日本無線㈱。

〈表3〉キャラクタ・コード一覧表

b <sub>3</sub> —b <sub>0</sub> \ b <sub>5</sub> —b <sub>4</sub>	0	1	2	3
0	A	N	0	↑
1	B	O	1	↓
2	C	P	2	←
3	D	Q	3	→
4	E	R	4	+
5	F	S	5	—
6	G	T	6	*
7	H	U	7	/
8	I	V	8	=
9	J	W	9	&
A	K	X	?	
B	L	Y	/	
C	M	Z	(アポストロフ)	
D	(ドット)	:	(ピリオド)	
E	—	■	□ (背景)	
F	(ブランク)	[	]	



## ビデオ・パターン・ジェネレータの製作

南雲敏行

ビデオ機器の調整などに欠かせないビデオ・パターン・ジェネレータを、専用ICにより製作してみます。

テレビ機器や画像処理システムの調整、修理そして性能の評価には、ビデオ信号のテスト・チャートを用いるのが一般的ですが、チューナを内蔵していないモニター・テレビの調整や、テレビ電波を受信できない場所・時間での受像機の調整には、ビデオ・パターン・ジェネレータが大変便利です。

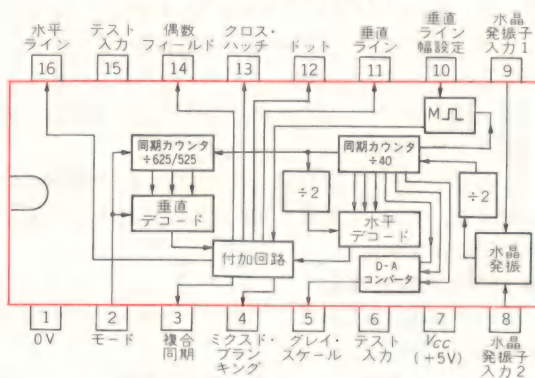
ここでは、1チップで簡単にクロスハッチ、ドット、グレー・スケールなどのテスト・パターンを発生することのできるIC ZNA234(フェランティ社、扱い；コーンズ・アンド・カンパニー・リミテッド、☎03-272-5771)、およびその応用について紹介します。

### ● ZNA234について

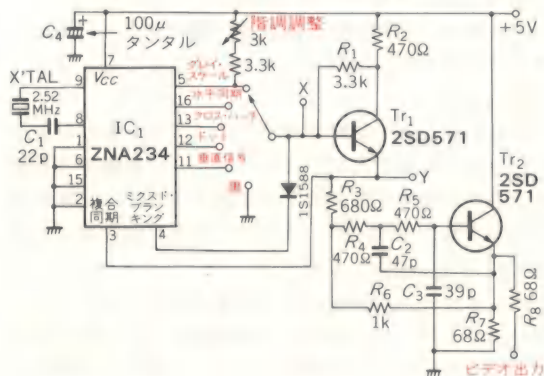
パターン・ジェネレータの応用範囲は広く、

- (1) 故障箇所の発見
- (2) 偏向回路の直線性調整
- (3) 受像機の感度試験

〈図1〉<sup>(4)</sup> ZNA234内部構成



〈図2〉パターン・ジェネレータ回路図



(4) 電波の出ていない時の各回路の調整

などがあります。

図1にZNA234の内部構成を、図2にパターン・ジェネレータの回路図を示します。必要な外付け部品は2.520MHzの水晶振動子と、少数の部品で簡単に構成することができます。

2.520MHzの水晶振動子は、安定度が重要でなければ約15pFのコンデンサで置き換えることもできます。また、外部クロックによって駆動することも可能で、この場合は9番端子を10kΩの抵抗でプルアップします(図3参照)。

### ● 水平同期信号を作るには

水平同期信号は、ミクスド・ブランキング出力を使って、ダイオード1S1588をONにし、Tr<sub>1</sub>をOFFにすることによって、まずブランキング期間(水平帰線時間)を作ります。次に、複合同期出力を使って、Tr<sub>2</sub>をOFFにすることによって、水平同期パルスを作ることができます。

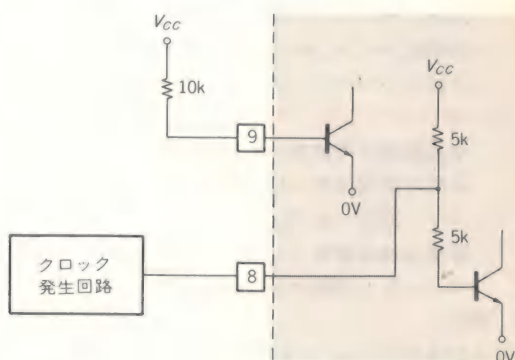
水平同期パルスの幅(0.08H±0.01H)は、ブランキング期間(0.16H~0.18H)より短く、水平同期パルスの前後はそれぞれフロント・ポーチおよびバック・ポーチと呼ばれます。また、Tr<sub>2</sub>の出力インピーダンスは低いので、出力インピーダンスを75Ωとするために、抵抗R<sub>8</sub>を接続してあります。

写真1に、このようにして端子5の階段状出力波形に水平同期パルスを作った波形を示します。

### ● 出力波形測定時の注意

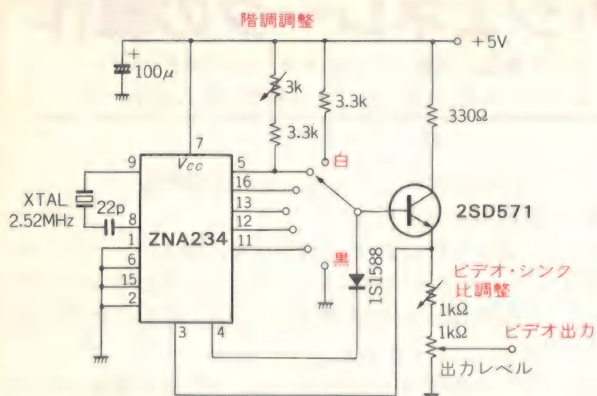
出力波形の測定を行う場合は、出力は75Ωで終端しなければなりません。C<sub>2</sub>とC<sub>3</sub>、R<sub>4</sub>そしてR<sub>5</sub>はローパス・フィルタを形成しており、高周波ノイズがビデオ

〈図3〉外部クロックによる駆動





〈図4〉 パターン・ジェネレータ回路B



出力に入るのを防いでいます。ビデオ・パターンをテレビの調整や修理のために使用し、送信機に接続することがなければこのフィルタは必要なく、回路を簡単にすることができます。この例を図4に示します。

また、ビデオ入力端子をもたないテレビなどで出力波形の測定を行うときには、**RFモジュレータ**(第3章参照)を使ってビデオ信号をRFに変換し、テレビのアンテナ端子から入力します。この場合も、ビデオ出力を75Ωで終端することを忘れてはいけません。

#### ● CCIRタイミングとEIAタイミング

ZNA234は、発振周波数およびモード入力の切り替えで、**走査線625本のCCIRタイミング**(国際電信電話諮問委員会によるもの、主にヨーロッパで適用)、または**525本のEIAタイミング**(米国電子工業会によるもの、主にアメリカ、日本で適用)で 사용할 ことができます。両者の各周波数および出力の違いを次に示します。

##### (a) 625ライン CCIRタイミング(モード=1)

水晶振動子周波数=2.50MHz

水平同期周波数=15.625kHz

ライン期間=64μs

垂直同期周波数=50Hz

フィールド期間=20ms

出力

- 20水平ライン(2ライン・ブランキング)
- 20垂直ライン(4ライン・ブランキング)
- クロスハッチ・アスペクト比 約1.4:1

##### (b) 525ライン EIAタイミング(モード=0)

水晶振動子周波数=2.520MHz

水平同期周波数=15.750kHz

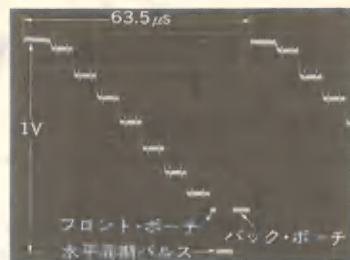
ライン期間=63.5μs

垂直同期周波数=60Hz

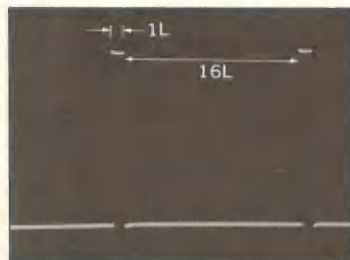
フィールド期間=16.66ms

出力

- 17水平ライン(2ライン・ブランキング)



〈写真1〉 コンポジット出力波形



〈写真2〉 水平ライン出力波形

- 20垂直ライン(4ライン・ブランキング)
- クロスハッチ・アスペクト比 約1.2:1

#### ● 水平/垂直ライン出力

水平ライン出力の波形(写真2)は、16ライン期間に1回発生する1ライン幅のパルスで、スクリーン上に2ライン幅(インタレースのため)の水平ラインを表示します。垂直ライン出力の波形(写真3)は、約3μsごとに生じる300nsの連続したパルスです。これらのパルスは、各ライン期間の同じ場所に生じるため、スクリーン上の垂直ラインとなります。

この二つの出力は、IC内部のANDおよびOR回路で処理され、ドット(写真4)およびクロスハッチ出力(写真5)となります。

#### ● テレビの調整方法

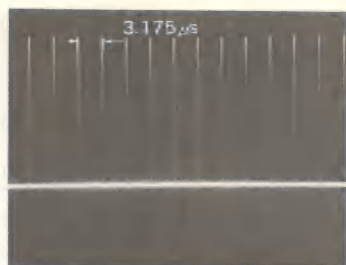
それでは次に、これらのパターンを使ったテレビの調整方法について述べます。

水平および垂直ラインの表示画面を写真6および写真7に示します。これらのパターンからは、画面全体の傾き、および水平・垂直直線性などを判断することができます。画面全体の傾きは、偏向ヨークの締めつけねじをゆるめ、偏向ヨークを回転させることによって調整します。

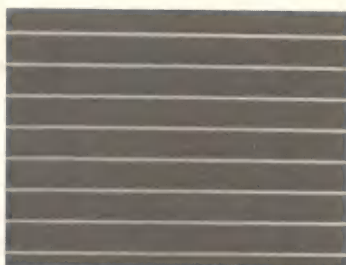
垂直および水平振幅が異常であったり、画面の上下が伸びたり縮んだりするという症状は、垂直偏向回路、および水平偏向回路の異常が原因ですので、正常な振幅ののこぎり波電流が、垂直および水平偏向コイルに流れるように調整します。

クロスハッチの表示画面を写真8に示します。クロスハッチでは、主に画面全体が樽状になる糸巻ひずみの補正を行います。糸巻ひずみは、糸巻ひずみ補正回





〈写真3〉 垂直ライン出力波形



〈写真6〉 水平ライン表示画面



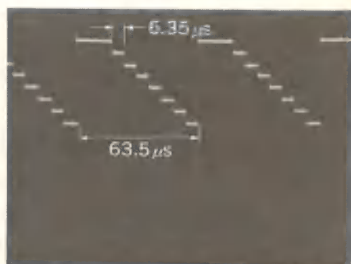
〈写真9〉 ドット表示画面



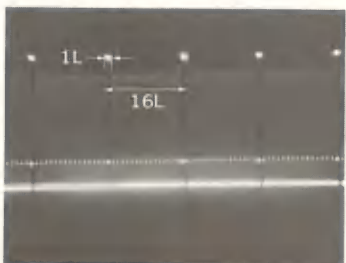
〈写真4〉 ドット出力波形



〈写真7〉 垂直ライン表示画面



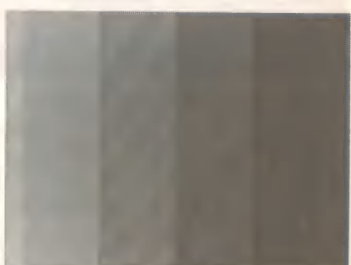
〈写真10〉 グレー・スケール出力波形



〈写真5〉 クロスハッチ出力波形



〈写真8〉 クロスハッチ表示画面



〈写真11〉 グレー・スケール表示画面

路、またはピンクッション・マグネットによって調整することができます。

ドットの表示画面を写真9に示します。ドットではフォーカスの調整を行います。また、カラー・テレビにこのドットを表示した場合、カラー・テレビのスタティック・コンバーゼンス(画面中央部の色ずれ)の調整を行うことができます。

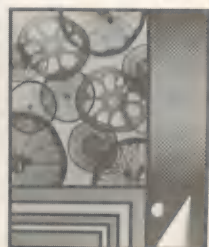
グレー・スケールの出力波形を写真10に、表示画面を写真11に示します。グレー・スケールは、正確な階調特性を得るために使用します。グレー・スケールは、

どちらかという、テレビ送信局で撮像デバイスの $\gamma$ 特性(階調特性)や、カメラの白・黒バランスの調整などに用いられることが多いようです。

#### ●参考・引用文献●

- (1) 平沢進, 宮崎直道, 細田祐造: テレビの調整と故障修理,
- (2) 吉野宏二: カラーTV修理マニュアル,
- (3) テレビジョン学会; テレビジョン画像の評価技術,
- (4)\*FERRANTI Integrated Circuits Data Book, コーンズ・アンド・カンパニー・リミテッド,

デジタルIC回路の設計  
湯山俊夫 著



## CORE BOOKS 第三弾

デジタルICの使い方が実感としてつかめます。

# デジタルIC回路の設計

実験で学ぶTTL, C-MOSの応用テクニック

湯山俊夫 著 A5判・256頁・定価1,650円

CQ出版社



## NTSC信号のA-D変換技術



尾津加 河湖

ディジタル画像処理に欠かせないビデオ信号のA-D変換には、NTSC信号特有の性質を理解する必要があります。ここでは、そのポイントとなる前処理回路技術について紹介します。

ディジタル画像処理、ディジタル・テレビなどの普及を技術的にささえているのが、ビデオA-Dコンバータと呼ばれるものの低価格化でしょう。

最近ではコラムでも紹介しているように8ビット、20Mサンプル/secクラスのA-Dコンバータが国産で安く入手できるようになってきましたので、この分野の発展はますます進むものと思われます。

### A-D変換のための前処理

#### ● サンプル周波数の決め方

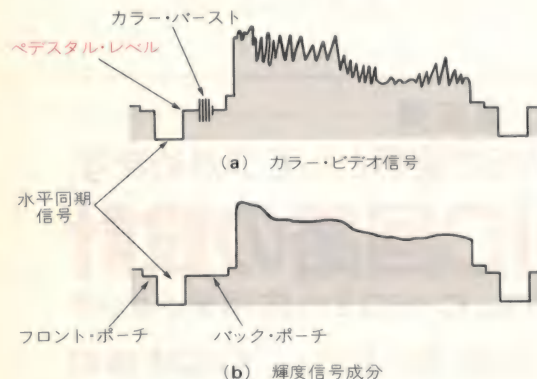
A-Dコンバータの扱うことのできる入力信号の周波数は、理論的にはナイキスト周波数(情報が欠落することなく再現できる最高周波数、)つまり**標準化周波数(サンプリング周波数)の1/2まで**です。しかし、現実には余裕をもって使用することが多く、通常NTSCコンポジット信号の変換には**色副搬送波周波数( $f_{sc}=3.579545\text{MHz}$ )の4倍の周波数**を使用します。

つまりNTSC方式を例にとると、

$$3.579545\text{MHz} \times 4 = 14.31818\text{MHz}$$

となりますから、15MHz以上のサンプリング・レートがあればビデオ用A-Dコンバータの資格があるといえます。現実に出回っている製品をみると、標準化周波数**20MHz前後**が主流となっています。

〈図1〉 走査線1本分の信号



#### ● クランプ回路

NTSCコンポジット信号をA-D変換するには、コンポジット信号の中から**輝度信号成分だけを**取り出し、さらにA-D変換に適した信号に加工する前処理が必要で、その一つがクランプ回路です。

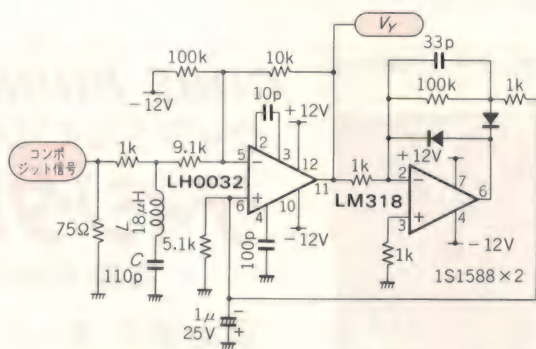
クランプ回路は、入力されたNTSCコンポジット信号の**DCレベルの変動を抑え、DCレベルを一定にする**ための回路です。ビデオ信号はオーディオ信号とは異なり、映像の明暗や色の情報、さらに同期信号が一つの信号として送り出されています。テレビ画面の走査線1本分の信号を図1に示しましょう。

この映像信号の輝度信号成分だけを見ると〔図(b)〕、**同期パルスの立ち下がるレベルがペデスタル・レベル**と呼ばれ、さらにその少し上に輝度成分の黒レベルがあります。

ビデオ信号をA-D変換するときには、このレベルを常に**A-Dコンバータの黒レベル**に保っておかなければなりません。そのため、同期パルスの底をクランプ・レベルとして一定に保ち(**シンク・チップ・クランプ**という)、黒レベルを安定させています。

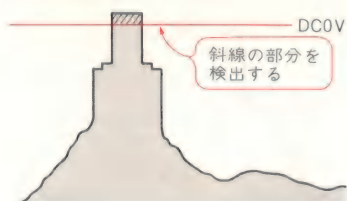
図2にクランプ回路の実際を示します。ここで使っているクランプ回路は、前段のビデオ・アンプが反転増幅になっていますから、この回路の出力信号は、グ라운드・レベルから負の方向に向かうビデオ信号を

〈図2〉 クランプ回路

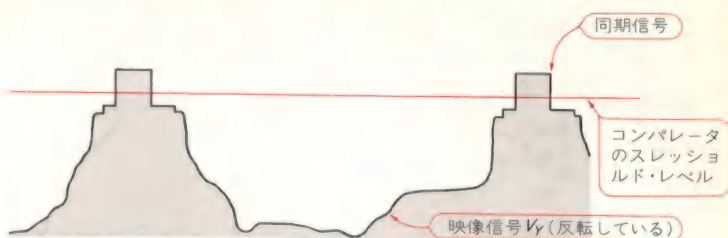




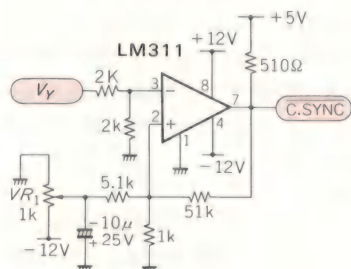
〈図3〉 クランプ回路のレベル変動検出



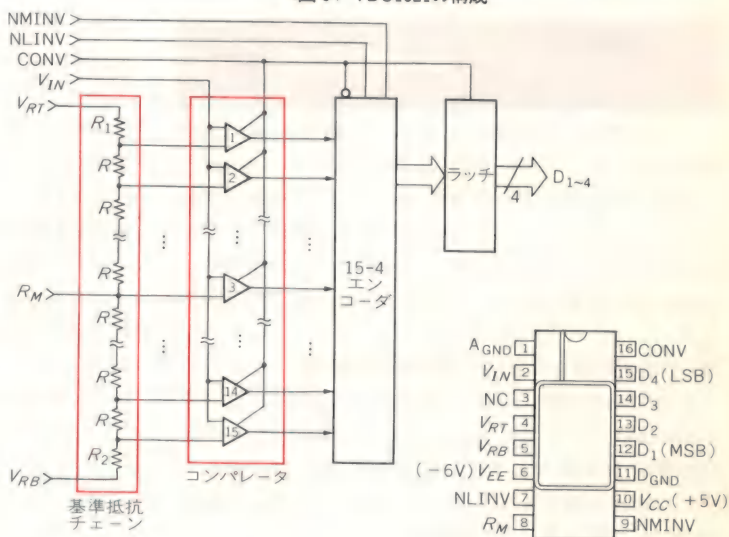
〈図4〉 同期分離の方法



〈図5〉 同期分離回路  
(ブランク信号回路も同じ)



〈図6〉 TDC1021の構成



出力します。この信号は一般のビデオA-Dコンバータの入力範囲(0～2V)に適合しています。

回路動作としては、前段で反転されたビデオ信号を次段で半波整流します。それにより図3のようにグラウンド・レベルより上に出た同期パルス部分を検出することになります。

このグラウンド・レベルより上に出た同期パルスを整流・平滑化して、前段の非反転入力にフィードバックしていますから、出力ビデオ信号の同期の底は、常にグラウンド・レベルに保たれることになります。このようなクランプ方式をフィードバック・クランプと呼びます。

この回路の動作の確認は、オシロスコープをDCモードにして、前段アンプの出力信号をモニタします。入力がビデオ・カメラの場合は、レンズの前を手でふさいだり、離したりしても、オシロスコープのビデオ波形の同期レベルが変動せず、常にグラウンド・レベルにあることを確認します。

VTRやテレビ・チューナの場合は、映像のシーンが変わってもDCレベルが変動しないことを確認します。

前段アンプの入力についているLCは、サブキャリ

ア除去フィルタで、ビデオ信号中の3.58MHzのサブキャリア成分を取り除きます。

LCの値は次式により求めます。

$$LC = \left( \frac{1}{2\pi f_c} \right)^2$$

$$f_c : 3.58\text{MHz}$$

Lはマイクロ・インダクタ、Cはスチロール・コンデンサ、またはセラミック・コンデンサを使用します。

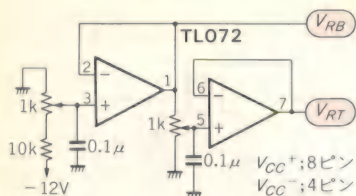
### ● 同期分離

次にビデオ信号から同期信号を分離して、TTLレベルの信号に直します。前段のクランプ回路で、同期信号の底は一定のレベルに固定されますから、図4に示すようにコンパレータで同期パルスの中央部をスレッシュホールド・レベルとしてスライスしてやれば、同期成分だけを取り出すことができます。

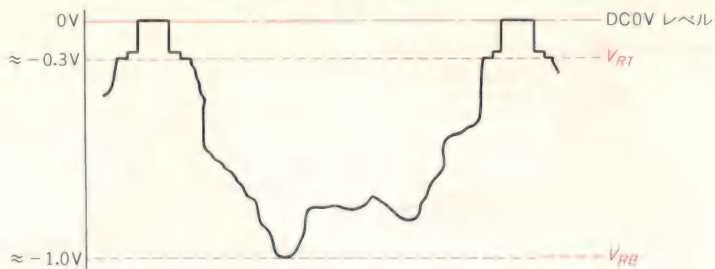
同期分離は、図5のようにコンパレータLM311により行います。前述のように同期パルスはグラウンド・レベルから負の方向に約0.3Vの範囲にきますから、オシロスコープでLM311の正負入力を見ながら、V\_R1で最も安定して同期分離が行われる値にセットします。



〈図7〉  $V_{RT}$ ,  $V_{RB}$ の発生回路



〈図8〉  $V_{RT}$ ,  $V_{RB}$ の調整方法



## A-Dコンバータの周辺回路

A-D変換は、NTSCコンポジット信号の中から輝度成分だけをデジタル信号に変換します。

通常、コンポジット信号をきれいにデータ処理するためには、6ビット以上のビット数を必要としますが、第8章で紹介するビデオ・エフェクタのような特殊な映像効果を得るには、4ビットくらいのほうが効果的です。

### ● 4ビットA-DコンバータTDC1021の例

ここでは代表例として、第8章でも使用しているTRW社の4ビットA-Dコンバータ TDC1021を例にして使い方を紹介します。図6がTDC1021の構成です。このICは最大25Mサンプル/secでの変換が可能です。ビデオ用としては最適です。

A-Dコンバータ TDC1021のアナログ信号入力範囲

は  $0V \sim -2.1V$  です。したがって、図2のクランプ回路の出力をそのまま入力することができます。ただし、クランプ出力は  $0V \sim -0.3V$  までは同期パルスであり、この部分はA-D変換する必要はありません。輝度信号の黒レベルから白のピーク・レベルまでをA-D変換するように設定します。

### ● 基準電圧の与え方

レベル設定のための機能として、TDC1021では  $V_{RT}$  と  $V_{RB}$  という二つの電圧入力端子があり、A-D変換する入力電圧範囲を次の条件で設定することができます。

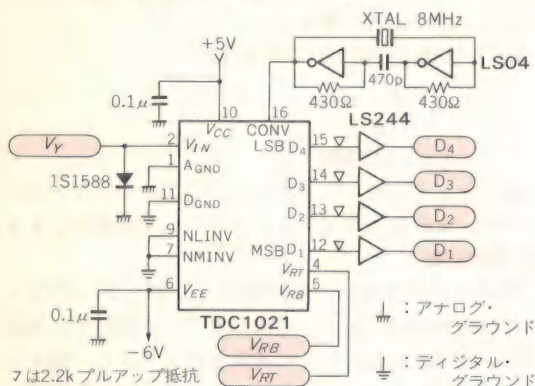
$V_{RT} \geq V_{IN} \geq V_{RB}$  の範囲で、

$+0.1V > V_{RT} > -1.9V$

$-0.1V > V_{RB} > -2.1V$

クランプ回路の出力信号  $V_Y$  は  $0V \sim -1.0V$  なので、 $V_{RT}$ ,  $V_{RB}$  ともにビデオ信号(輝度信号)の全レベル範囲に設定することが可能です。

〈図9〉 A-Dコンバータ周辺回路



〈表1〉 A-Dコンバータ出力コーディング

ステップ	-1.0V FS 66.667mVステップ	NMINV="1" NLINV="1"	NMINV="0" NLINV="0"
0	0.000	0000	1111
1	-0.067	0001	1110
2	-0.133	0010	1101
3	-0.200	0011	1100
4	-0.267	0100	1011
5	-0.333	0101	1010
6	-0.400	0110	1001
7	-0.467	0111	1000
8	-0.533	1000	0111
9	-0.600	1001	0110
10	-0.667	1010	0101
11	-0.733	1011	0100
12	-0.800	1100	0011
13	-0.867	1101	0010
14	-0.933	1110	0001
15	-1.000	1111	0000

〈図10〉 A-Dコンバータのサンプリング・タイミング





図7により、この二つの基準電圧を設定します。 $V_{RB}$ をホワイト・ピークに、 $V_{RT}$ を黒レベルに設定します。ここでは、

$$V_{RB} = -0.3V, V_{RT} = -1.2V$$

とします。オシロスコープでクランプ出力を見ながら図8のように調整します。図9がA-Dコンバータ周辺の回路図です。

CONV入力は、A-D変換のタイミングを与えるものです。フラッシュ型と呼ばれるTDC1021では、図10に示すように、CONV信号の立ち上がりから約10

ns後にサンプリングが行われ、そのデータは次の次の立ち上がりエッジで出力されます。

A-Dコンバータにより変換されるビデオ信号レベルと、出力データの関係は表1のようになります。

この表から、 $NMINV = NLINV = "1"$  のときと、 $NMINV = NLINV = "0"$  のときには、出力データが反転することがわかります。これを利用して、出力の映像をネガ反転させることができます。これは第8章で紹介するビデオ・エフェクタなどでは有効に使えるものです。

## 国産ビデオA-Dコンバータ

軍用や産業用電子機器を主な市場とし、ほんの数年前までは10万円以下では入手できなかったビデオ帯域のA-Dコンバータが、半導体技術の進歩により大幅にコストが下がりました。

最近では民生機器でもデジタル画像処理技術を取り入れた例（VTRのトリック・プレイなど）が増えています。したがって、その潜在需要は大きく、近頃では国内ICメーカーがいっせいにこの市場をねらってA-Dコンバータを開発、メーカーによっては量産出荷を開始しています。

こうなってくると、価格は加速度的に下がります。サンプルでも数千円、量産時には千円弱まで下がると思われます。すると各種応用範囲はさらに拡大し、これからの8ビット20Mサンプル/secクラスのA-Dコンバータはおもしろくなりそうです。

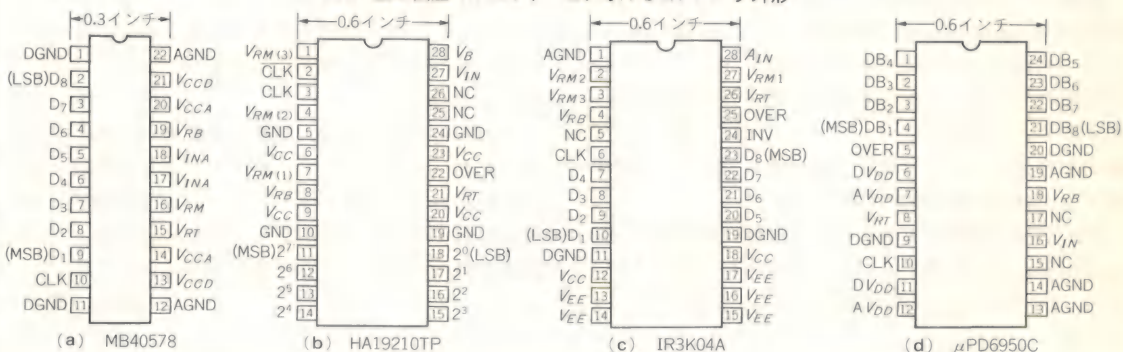
写真Aが、国産メーカーから出されているビデオA-Dコンバータの一例で、図Aにその外形図、表Aに電気的特性を要約したものを示します。メーカーとしては、このほかにソニー、松下電子、東芝などがありますが、ここでは出力がTTL型になっているものだけを取り上げました。ほかのものはECL出力

〈写真A〉  
国産の8ビット・  
ビデオA-Dコンバータ



になっているものが多いようです。20Mサンプル/sec以上のデバイスになるとECL出力というのは必然的になりますが、手軽に使うという点ではTTL出力型

〈図A〉 主な国産の8ビット・ビデオA-Dコンバータ外形

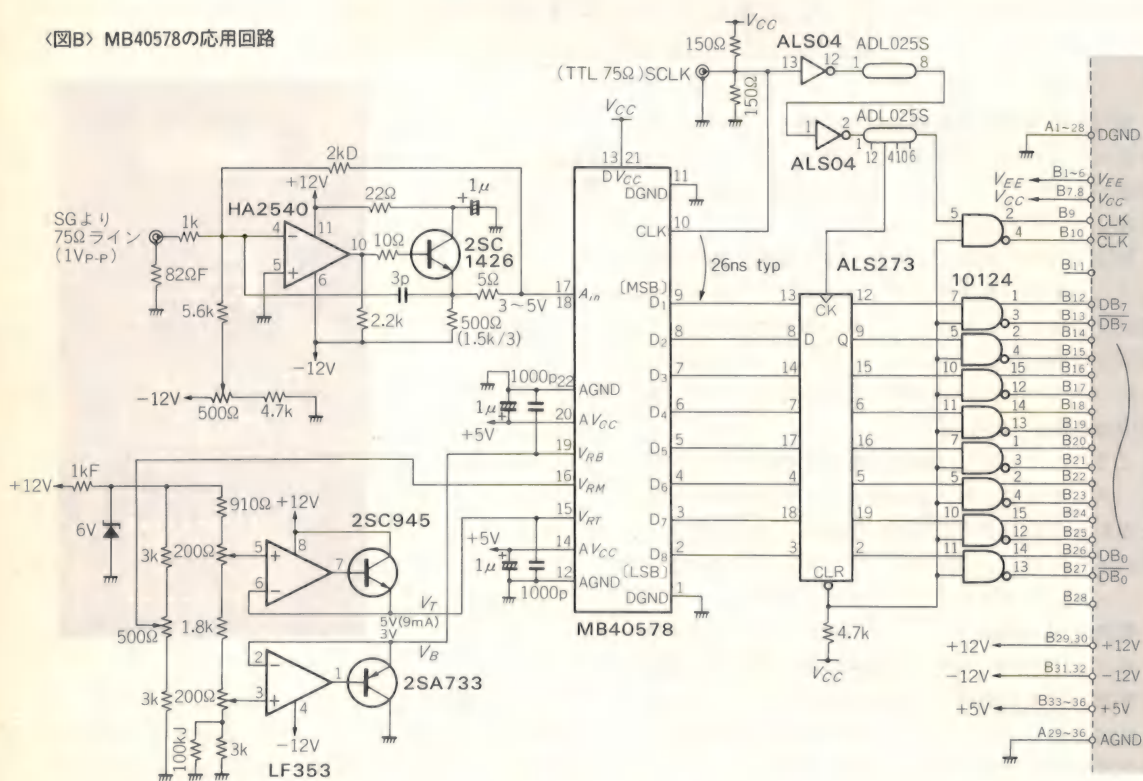




〈表A〉 国産の8ビット・ビデオ  
A-Dコンバータ  
(TTL出力型のみ取りあげた)

型 名	MB40578	HA19210TP	$\mu$ PD6950C	IR3K04A
メーカ	富士通	日立	日本電気	シャープ
分解能(ビット)	8	8	8	8
プロセス	バイポーラ	バイポーラ	C-MOS	バイポーラ
寸法	0.3インチ, 22ピン	0.6インチ, 28ピン	0.6インチ, 24ピン	0.6インチ, 28ピン
直線性精度	$\pm 0.2\%$	$\pm 0.5\text{LSB}$	$\pm 1.5\text{LSB}$	$\pm 0.5\text{LSB}$
変換速度	DC~30MHz	DC~30MHz	DC~20MHz	DC~20MHz
アナログ入力 電圧範囲	3.0~5.0V	1.5~3.5V	0~3.5V	0~-2.0V
ディジタル 入出力電圧	TTLレベル	TTLレベル	TTLレベル	TTLレベル
電源電圧	+5V単一	+5V単一	+5V単一	+5V, -7V
消費電力	480mW	250mW	400mW	1W

〈図B〉 MB40578の応用回路



のほうが有利です。

図Bが、代表的な国産ビデオA-DコンバータであるMB40578（富士通）の応用回路です。この種のA-Dコンバータは入力インピーダンスが高くなく、さらに入力電圧の大きさによっても値が変動する可能性がありますので、**入力段にはバッファ・アンプを用意する必要があります**。ここではビデオ帯域まで使用できるOPアンプ HA2540を使用しています。

安定なA-D変換を実現するには、基準電圧 $V_{RM}$ 、 $V_{RT}$ も安定なものがが必要です。ここでは少し高級ですが、 $V_{RM}$ 、 $V_{RT}$ を任意に設定できるようにした例を示しています。

表Aを見るとわかるように、大いこのA-Dコンバータは**5V単一電源で動作**するようになっています。したがって、周辺回路をうまく工夫すれば、システムとしての5V単一電源化も可能でしょう。工夫してみてください。

なお、この種のA-Dコンバータの使用実例としては第9章（ビデオ・フィールド・メモリの設計・製作）、第10章（倍速スキャン・コンバータの設計・製作）に $\mu$ PD6950C（日本電気）が出てきます。周辺回路の設計には、これらを参考にするのがよいでしょう。

〈魚田 隆〉



## 多機能ビデオ・エフェクタの 設計・製作



尾津加 河湖

ビデオ機器にエフェクタを入れると、様々な画像を楽しむことができ、画像処理の入門用としても最適です。ここでは疑似カラー、ディフェクト、モザイクなどの効果を実現する手法を紹介し、**巻頭のカラー口絵**も参照してください。

これまで一般ユーザ用ビデオ周辺機器というと、エンハンサやテロップ、カラー・コレクタといったものが主でしたが、ビデオ・カメラやビデオ・デッキの普及に伴い、ビデオ・エフェクタというものも何種類か出てきています。

ビデオ・エフェクタは、サウンド・エフェクタ(エコー、ボコーダ、フェーダなど)が音を加工するのに対して**映像を加工するもの**をいいます。

テロップやカラー・コレクタもエフェクタの一種といえますが、**ワイプ、クロマキー、ミラー効果、ディフェクト、ページめくり、モザイク**などが、テレビ放送やプロモーション・ビデオではよく見かけることができます。

最近のプロ用ビデオ・エフェクタはイメージ・メモリを使い、ほとんどディジタル処理によって特殊効果を得ています。ここでは簡単に作ることができて、しかも効果の大きいディフェクトと疑似カラーを中心に、ワイプ、フェーダ、スリット、シフト、モザイク、スライスなどの特殊効果を得るための回路の作り方を説明したいと思います。

### エフェクタの基本構成 ——ディフェクト&疑似カラー——

今回製作するビデオ・エフェクタの基本となるのが、ディフェクトおよび疑似カラーと呼ばれるものです。まず、この効果から説明しておくことにしましょう。

#### ● ディフェクト

少しページをもどりますが**巻頭の口絵ページ**を見てください。この中の**写真2**は最近のスポーツ番組やニュース番組の頭で、このような映像を見かけることが多いと思います。この効果はディフェクトと呼ばれ、専用のエフェクタもありましたが、比較的簡単な回路でこの効果を作り出すことができます。

原理的にはビデオ信号をA-D変換してディジタル化し、それを再びD-A変換します。そして、**元の映像信号を再構成する際にビット数を少し減らしてやればよい**のです。それによって、油絵で描かれたような映像が得られます。

またD-Aコンバータに送るデータの**ビットの選び方をいろいろ変えてやると**、口絵の**写真4**のような**ソラリゼーション効果やネガ反転映像**が得られます。

これらの回路は、ディジタル・ビデオ・エフェクタの最も簡単な、また基本的な回路といえます。また、A-DコンバータとD-Aコンバータの間に、メモリを入れたり演算回路を入れたりすることで、様々なビデオ・エフェクタへと発展させることができます。

#### ● 疑似カラー

疑似カラーの最も一般的な例は、コンピュータ・グラフィックスでしょう。

コンピュータ・グラフィックスでは、グラフィック・データのビットに色を割り当てて、RGBモニタに映像信号として出力しています。

それと同じように、ビデオ映像信号の輝度信号成分をA-D変換して何ビットかのデータを得ます。そして、この**データのビットにRGB成分としての色を与え、それをRGBモニタに出力**します。すると、元の映像の色とは**まったく関係ない色付けをされた映像**が得られます。これが疑似カラーです(口絵の**写真7**参照)。

疑似カラーは、元の映像信号の輝度信号レベル(電圧)に応じた色を付けますから、これを利用してサーモグラフ(温度は赤外線テレビ・カメラの輝度に相当する)などが作られています。これは科学番組などでよく見かけるといいます。

#### ● 基本回路の構成について

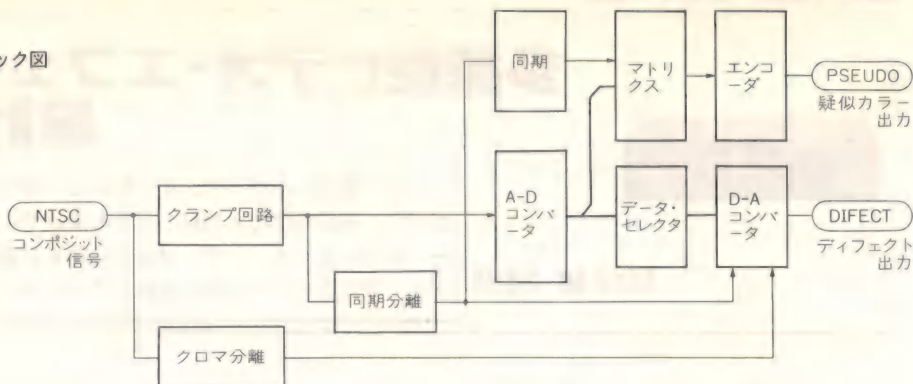
図1にエフェクタのブロック図を示します。回路はシンプルです。しかも調整部分も少ないので、回路図どおりに作れば確実に動作します。また、この回路を基本回路として様々なビデオ映像装置に発展させることができます。

実際の回路図は図2のとおりです。以下、各部分の設計法、注意点について説明します。

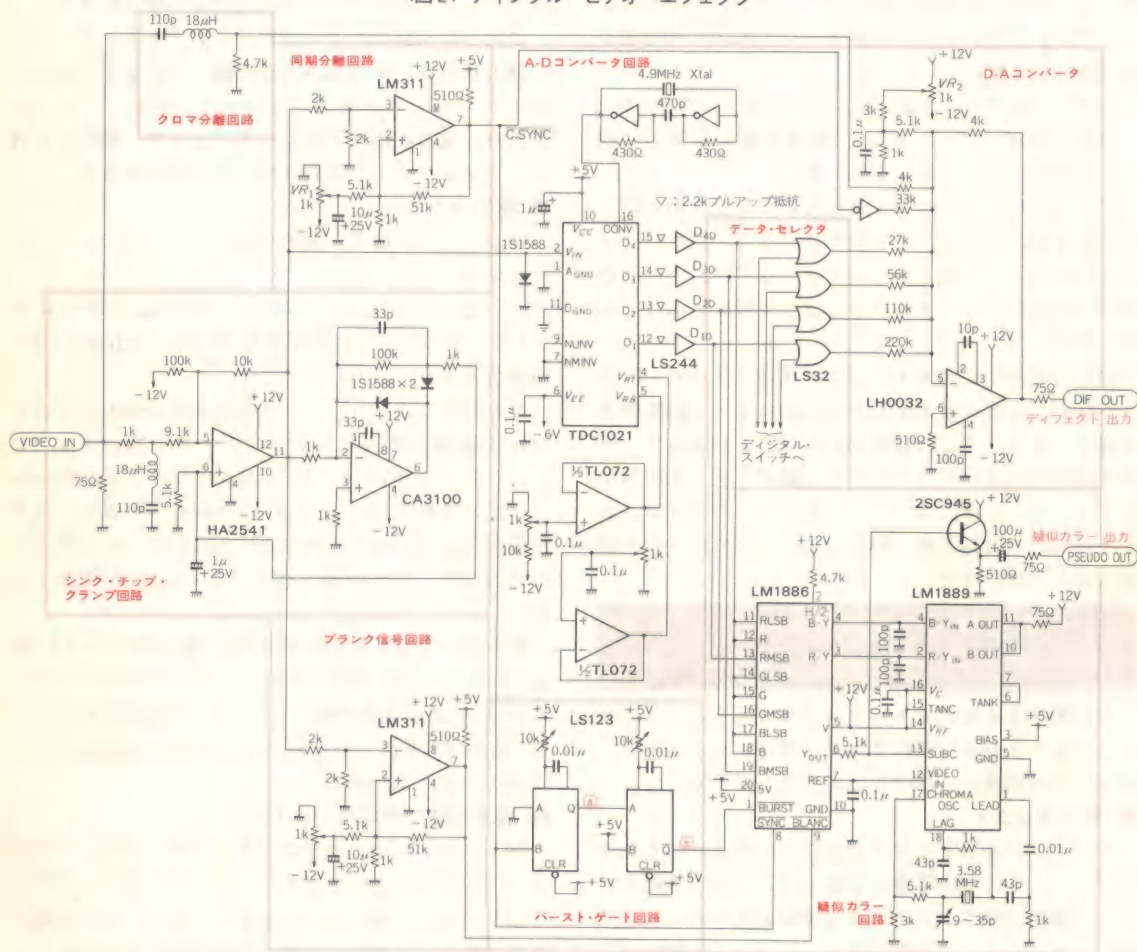
なおA-D変換されたデータそのままでも、RGBモニタで見ることができますが、VTRに録画することはできません。したがって**録画したり一般のテレビで見るようにするには、これをNTSCコンボジット信号**



〈図1〉  
ビデオ・エフェクタのブロック図



〈図2〉 デジタル・ビデオ・エフェクタ



に直してやらなければなりません。これには専用のエンコーダICを使用して変換します。

エンコードする際にはどうしても解像度が少し落ちますから、画像処理や画像計測で高い解像度が必要な場合は、RGBモニタで確認したほうが良いでしょう。

#### ● NTSCコンポジット信号のA-D変換

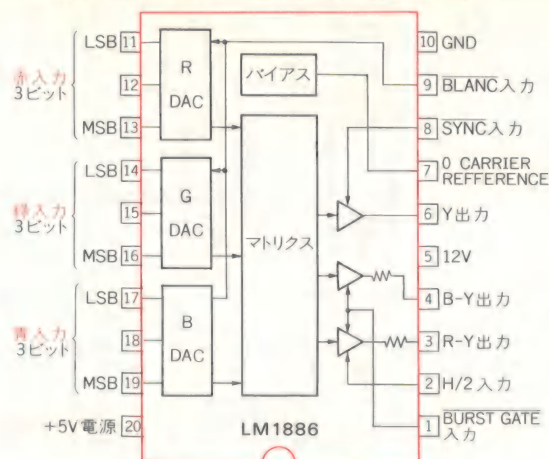
各種の映像効果を作るには、コンポジット信号をA

-D変換してからデジタル的に操作します。コンポジット信号をA-D変換するには、シンク・チップ・クランプや同期分離などの前処理が必要ですが、これらの回路は第7章で紹介したとおりです。

ここでは4ビットのA-Dコンバータ TDC1021を使用しますが、TDC1014を使用すればこのままの回路で6ビットのA-D変換が可能になります。



〈図3〉 テレビ・ビデオ・マトリクスD-Aコンバータ



なお、図1においてクロマ分離の部分は、ビデオ信号のうちの色信号(クロマ信号)を分離する回路ですが、今回は色信号自体は加工しないので、一度分離しておいて最後に再びビデオ信号にもどしてやります。

#### ● データ・セレクトの構成

A-D変換して得た4ビットのデータを、ディフェクト(欠乏)させるための回路です。最上位ビットと最下位ビットだけを選択するというソラリゼーションの効果が得られます。また最上位ビットだけを選択すると2値画像(口絵の写真6)が得られます。そしてデータ・セレクトで選択されたビットの信号をD-A変換し、映像信号に再構成します。

このとき先に分離したクロマ信号と同期信号をミキシングして、NTSCコンポジット信号として出力させます。この信号は通常のビデオ・モニタやVTRに入力することが可能です。

また、ここでクロマ信号をミキシングさせずに出力すると、出力信号は白黒映像となり、これも特殊効果として使用することができます。

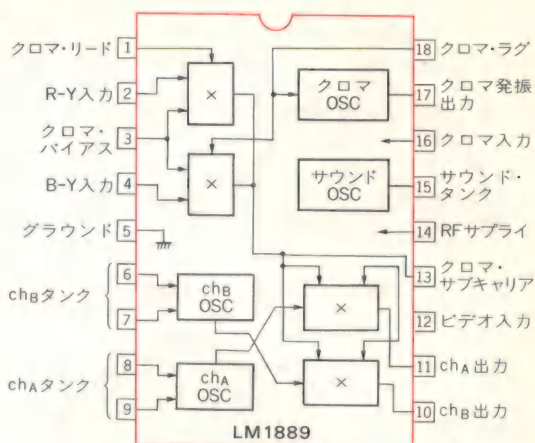
#### ● D-Aコンバータの構成

D-Aコンバータはビデオ専用のICもありますが、6ビット以下なら抵抗とOPアンプによる構成でも十分です。D-A変換は、 $D_{1D} \sim D_{4D}$ までの4ビット・データをアナログ値に変換します。各抵抗値は第7章の表1にしたがって、全ビットが“H”のとき変換値が0.7Vになるようにします。

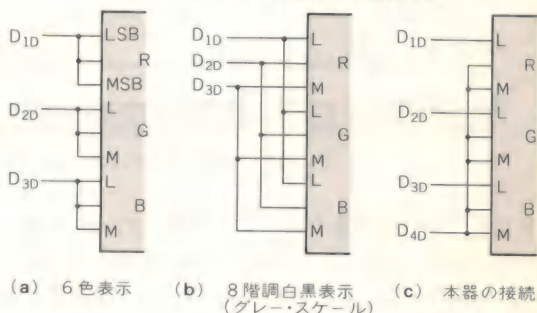
同様に、TTLレベルで入ってくる同期成分が0.3Vになるように抵抗を選びます。もう一つ、4kΩの抵抗を通して入ってくるのは、クロマ成分です。先のサブキャリア・フィルタと同じ定数にセットされたフィルタにより、3.58MHzのサブキャリア成分のみを取り出し、それをこのミキシング回路に加えています。

出力ビデオ信号のクロマ成分があまり小さすぎる場

〈図4〉 テレビ・ビデオ・モジュレータ



〈図5〉 ビデオ・マトリクスの結線方法



合、あるいは大きすぎる場合は、この4kΩの抵抗で調整します。

ディフェクト効果の回路構成はこれだけですが、A-Dコンバータの $V_{RB}$ 、 $V_{RT}$ を条件の範囲内で変化させると、フェーダ的な使い方も実現できます。

#### ● 疑似カラー・エンコーダ

ここではLM1886(ビデオ・マトリクスDAC)と、LM1889(ビデオ・モジュレータ)の二つのICを使って疑似カラー出力を作っています。

図3にLM1886、図4にLM1889の構成を示します。

LM1886はビデオ・マトリクスD-Aコンバータで、RGB各3ビットの入力データから、色差信号と輝度信号をエンコードするものです。

輝度信号出力はNTSC方式で、

$$Y = 0.3R + 0.59G + 0.11B$$

の式によりエンコードされます。また、R-Y、B-Y出力は過変調を防ぐようにウェイトされています。

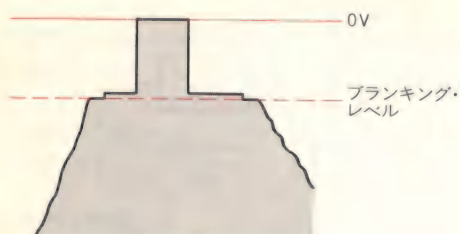
LM1889はLM1886とペアで使用することで、LM1886の色差信号をコンポジット信号として出力することができます。

このICペアは、サーモグラフや魚群探知機などの疑似カラー表示回路としてよく利用されています。

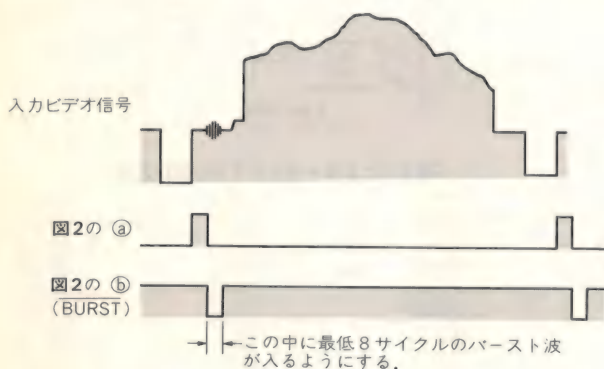
RGB入力信号はR、G、Bそれぞれ3ビットずつ入



〈図6〉ブランキング信号スライス・レベル



〈図7〉バースト・ゲート信号



力することができますが、ここで使っているA-Dコンバータは、4ビットのものを1個しか使っていません。したがって、4ビットをこの9ビットの入力に振り分けなければなりません。

図5(a)のように配線すれば6色の表示になり、図(b)のように配線すると8階調の白黒映像が得られます。ここでは図(c)のように配線しますが、この組み合わせは任意に変えることができますから、自由に変えて効果を試してください。

### ● ブランク信号とバースト・ゲート信号を作る

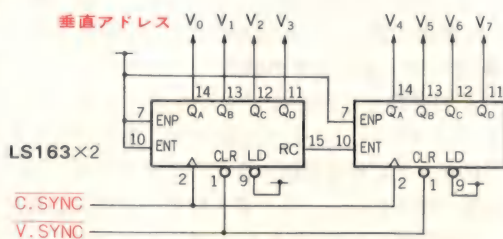
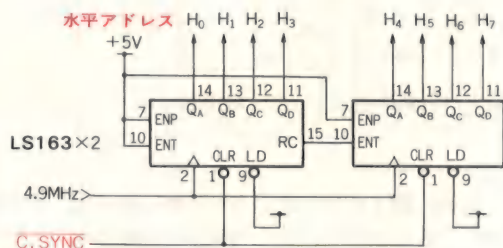
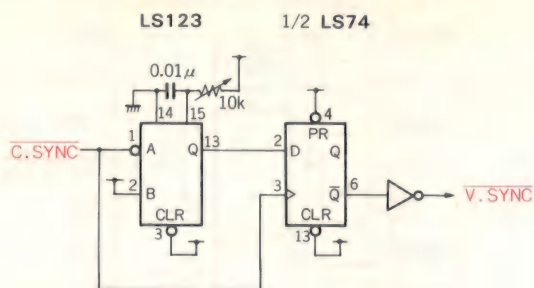
ブランク信号は簡易的な方法により作ります。同期分離の項で説明した方法とまったく同じ方法です。スライス・レベルを図6に示すようにペダスタル・レベルよりもわずかに上に設定します。この方法でブランキング信号を簡単に得ることができます。

しかし、この方法では入力ビデオ信号の質が悪い場合にジッタが現れることがあります。本格的には先に分離したコンポジット・シンクとワンショット・マルチバイブレータによって作ります。

バースト・ゲート信号は、二つのワンショット・マルチバイブレータにより、コンポジット・シンクから作ります。この信号は、カラー変調信号の基準位相を示すバースト信号をビデオ信号(のバック・ポーチ)に乘せるための位置を決める信号です。この位置は、入力ビデオ信号のバースト位置を見ながら図7のようにオシロスコープで調整するとよいでしょう。

クリスタルは3.579545MHz、一般に3.58MHzとい

〈図8〉付加のためのドライブ回路



われる周波数のものを使用します。発振が正しく行われていると、LM1889の1番ピンと18番ピンに現れる発振波形は互いに90°位相が異なる波形(約500mV<sub>P-P</sub>)が得られます。

簡単な回路なのでほとんど調整する所はありませんが、出力ビデオ信号が白黒であったり、またモニタ上ではカラーなのにVTRに録画すると白黒になってしまうような場合は、発振回路を調整してきれいなサイン波が得られるようにしてください。

### ● 設計上の基本的な注意点

回路製作上の注意すべき点は、まず基板はガラス・エポキシのものを使用し、パターン化する場合、いわゆるベタ・アースを基本として、各パターンは太めに設計するとよいでしょう。

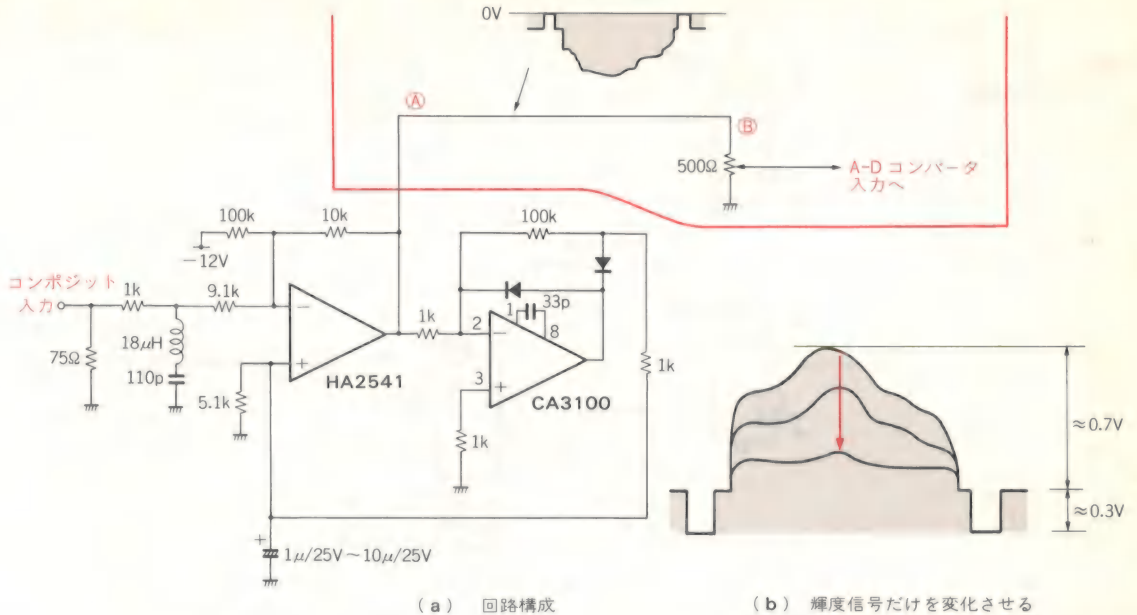
また、グラウンドは、デジタル系とアナログ系を分けて配線し、電源端子のところで1点アースとなるようにします。

クロックの信号線は引き回さずに、発振部と入力する素子とはできるだけ近くに配置するようにします。

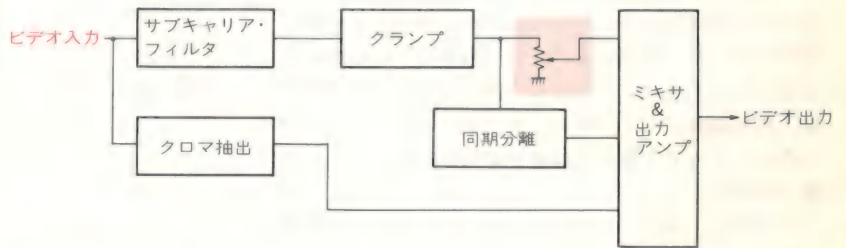
発生するノイズとしては、クロック成分のクロストークが最も発生しやすいので、A-Dコンバータの周辺は特に注意してください。



〈図9〉 フェーダの追加



〈図10〉  
ディフェクトの回路を使  
わずにフェードするには



## エフェクタへの各種付加回路

### ● ドライブ回路の追加

前述のディフェクトと疑似カラーでは、A-DコンバータとD-Aコンバータの間にはほとんど回路といえるものは入っていません。しかし、この間にエフェクト回路を入れることにより、様々な効果を得ることができます。

エフェクト回路を入れる前に多少の追加回路を作っておきます。これはエフェクト回路をドライブするのに必要なもので、図8に示すようにC. SYNCからLS123とLS74でV. SYNCを作ります。また、カウンタによって、C. SYNCとクロックから水平アドレス、V. SYNCとC. SYNCから垂直アドレスを作ります。ここでは水平、垂直を256×256画素に設定しています。回路は少し大き目の基板に一つずつ動作を確認しながら作っていくとよいでしょう。

以下、エフェクトは、その基本機能だけを説明しま

すから自由に変形させて効果を楽しんでください。

### ● フェーダ

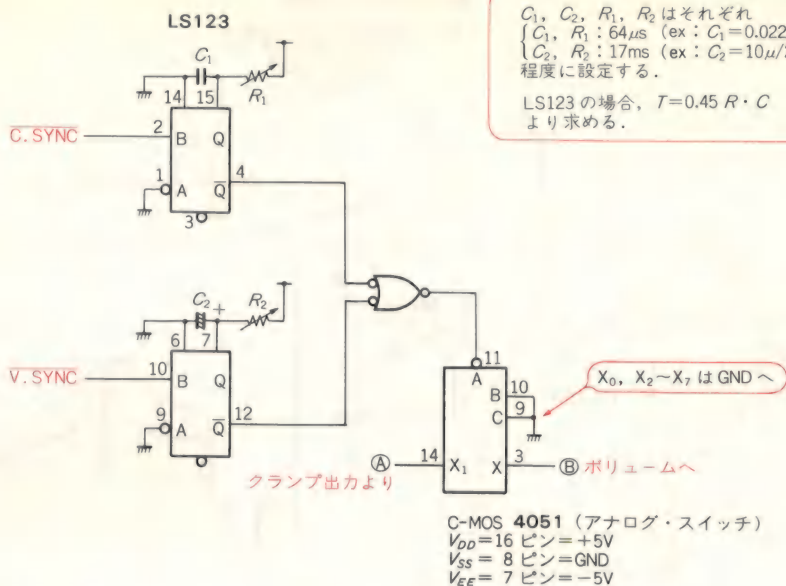
フェーダは、オーディオ・ミキサのフェーダと同様に、フェード・イン、フェード・アウトの機能です。しかし、ビデオ信号の場合は同期成分が信号に含まれていますから、単純に減衰や増幅を行ったのでは、同期成分まで変化してしまうため映像は正常に映りません。

先にディフェクトの回路がありますから、この回路に図9のようにボリュームを1個挿入するだけでフェーダの機能をします。これにより画面が徐々に消えたり、また徐々に現れたりする効果が得られます。口絵ページの写真9と写真10にフェードの様子を示します。

フェーダでは、図(b)に示すように、同期信号はそのままで輝度信号成分だけを変化させなければなりません。また、カラー・ビデオ信号の場合には、カラー・バーストおよびサブキャリアが含まれていますから、これらは変化させずにフェードしなければなりません。そのため、ディフェクトの回路を使わずに、アナログ



〈図11〉  
ワイブを作る回路



のビデオ信号をそのままフェードするには、図10に示すような回路構成でなければなりません。

このやり方では正確には、輝度信号とサブキャリア成分では時間的なずれを生じますが、実用上それほど問題にならないと思います。もし色がにじむと思われる場合には、クランプ出力の映像信号にディレイ素子を入れて調整してください。ただし、このフェードでは白にフェードすることはできません。

### ● ワイブ

ワイブは、画面の一方あるいは複数の方向から映像を消していく手法で、回路的には様々な方法が考えられます。図11の方法は、水平同期(C. SYNCで代用)と垂直同期信号を使ってLS123でゲート信号を作り、アナログ・スイッチ(この場合は4051)をON/OFFさせることで映像を消去させています。

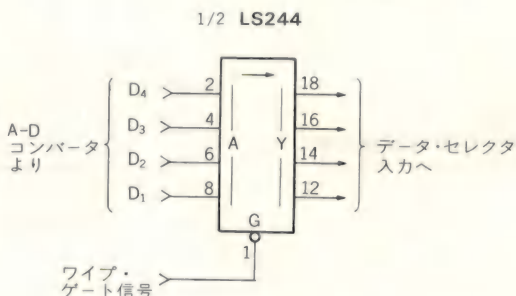
この回路は、ワンショットを使って水平同期の立ち上がりから(画面の左端から)、ある時定数のパルスを作り、その信号をゲート信号として利用しています。ワンショットの時定数はボリュームで調整しますが、このボリュームを引き出すときは、できるだけ短くなるように配置してください。

垂直側もまったく同様に、垂直同期信号からパルスを作ります。ここでは回路を簡略化していますから、ゲート・パルスの終わりは、きちんとラインの終わりにはなっていません。そのため、垂直側のワイブの切れ目は、左右に少しちらちらするかもしれません。

垂直側のゲート・パルスと水平側のゲート・パルスのORをとると、両方から同時にワイブをかけることができます。

また図12に示すように、A-Dコンバータの出力バ

〈図12〉 ワイブを作る別の方法(1)



ッファに使っているLS244のゲートに、この信号を入れても同様の効果が得られます。こちらの方法では、ワイブできるのは必ずA-D変換した信号になりますから、ディフェクトやシフト、スリット効果(後述)に使用できます。

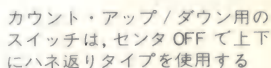
また、LS244のゲートGを“H”にすると出力はハイ・インピーダンス状態になりますから、そのとき出力側を抵抗でプルアップしておく、ゲートが“H”になったときに全出力信号が“H”になります。

図13に示すのは、アドレス信号を利用したもので、デジタル・コンパレータでワイブのゲート信号を作っています。この方法はデジタル処理ですので、外部から制御したり、複雑な処理も行えます。

この回路は、先に作っておいたアドレス信号をTTLのコンパレータ、LS682に入れ、もう一方のコンパレータ入力にはアップ/ダウン・カウンタの出力を入れてやります。アップ/ダウン・カウンタは、スイッチによってアップ/ダウンし、その値よりも小さい範囲をワイブさせます。



&lt;图13&gt;



&lt;图14&gt;



水平側と垂直側カウンタの、どちらを動かすかによってワイプの方向が異なります。図14にその例を示します。

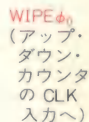
ワイブのスピードはカウンタのクロック入力によります。クロックは、図15に示すようなオシレータを用意するか、図16に示すように、アドレスの  $V_7(1/60 \text{ 秒})$  をカウンタで分周し、適当なスピードを選ぶようにします。口絵の写真11、写真12がワイブの実例です。

## ● スリット効果

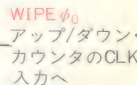
ここではスリット効果と仮に名付けましたが、効果としては口絵の**写真13**のように、**縦または横のスタレ**状の画像です。

基本的にはワイプと同じ考え方ですが、ここではアドレス信号をゲート信号として使用します。例えば、水平アドレスを見ると、1, 2, 3, …画素ごとの信号ですから、これをゲート信号として使えば1, 2, 3, …画素ごとに画像が現れることになります。

〈図15〉ワイプ用のクロック・オシレータ



〈図16〉ワイプのスピード可変用カウンタ

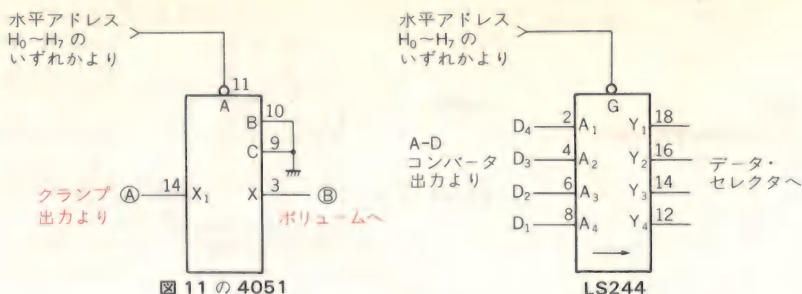


垂直方向も同様に何ラインごとかに画像が現れます。  
また、口絵の**写真14**は水平と垂直アドレスの**AND**をとって入力した例で、チェック模様の画像になります。

回路は図17に示すように、単純にアドレス信号をゲートに入れているだけです。しかし、これを図18に示すようにデータ・セレクトで水平、垂直のゲート信号を選択する回路を付加させると便利になります。

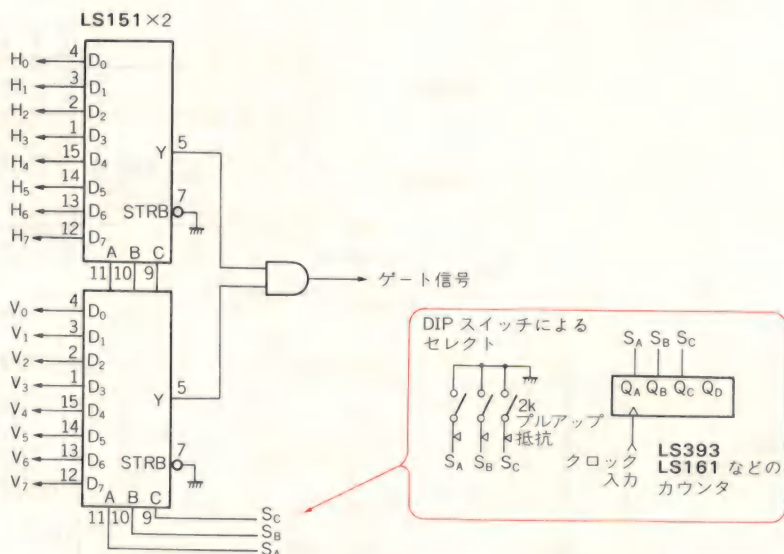


〈図17〉 スリット効果を作る



(a) アナログ・ゲートを使用した場合 (b) デジタル・ゲートを使用した場合

〈図18〉  
スリット効果をさらに  
上げるセレクト



また、データ・セクタのセレクト入力、ディップ・スイッチで簡単にセレクトするほかに、**カウンタ**をもう一つ用意して、**時間的に変化**させてやることもできます。これは**ダイナミックな効果**が得られます。また、セレクトするアドレス信号をランダムに並べ替えてやると、さらに変化のある効果が得られます。

### ● スライス効果

スリット効果ではアドレス信号でゲートをかけましたが、**ある輝度レベル以下の信号をカット**したらどうでしょうか。ある輝度レベルに関して、上か下かを示す信号は、コンパレータによって得られます。アナログのコンパレータでもかまわないのですが、ここでは図19のようにTTLを使ってコンパレートします。

そうすると口絵の**写真15**に示すように、**あるレベル以上(または以下)の部分**が黒く**カット**され、そのレベルでスライスしたように見えます。この黒くなった部分へは、他の映像信号をはめ込むことができます(ただし同期が一致していなければならない)。

この効果は、**クロマキー**と呼ばれる効果に利用できます。つまり、ゲート信号を得るために、スライスする映像信号を、例えばRGBデコードしたGの信号を使

うと、映像中の**緑の色の強い所だけがぬける**ことになります。したがって、緑の壁をバックに人物などを撮像すると、**人物だけが切りぬかれた**ようになります。

ここでは、輝度信号をスライスすることでゲート信号を作っていますから、画面の明るい部分または暗い部分がカットされることになります。

### ● レベル・シフト効果

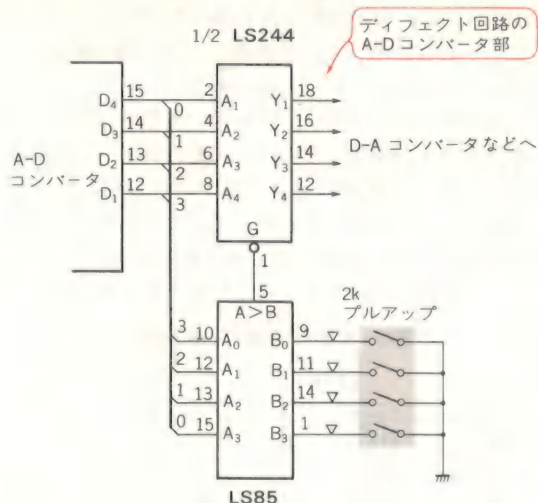
すべてのA-Dコンバータでできるわけではありませんが、ここで使っているTDC1021では、**基準電圧  $V_{RB}$ 、 $V_{RT}$  端子の電圧を広範囲(0〜2V)に変化**させることができます。したがって、この端子に図20のようにして定常時な電圧ではなく、別な信号を入れてやることで、A-D変換出力信号に変化を与えることができます。

$V_{RB}$ 、 $V_{RT}$ の端子電圧はディフェクトのところで説明したように、A-D変換する入力信号の範囲を決めています。

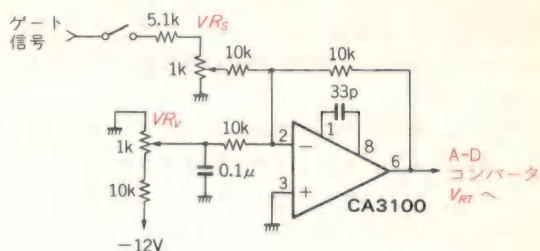
$V_{RT}$ ( $V_{RT}$ 端子のグラウンドに対する電位)はA-D変換**レベルの上限**、 $V_{RB}$ は**下限**を決めています。 $V_{RT}$ と $V_{RB}$ の差が小さいほど、入力信号のせまいレベル範囲を変換することになるので、これはゲインに相当しま



〈図19〉 スライス効果のための  
輝度レベルを検出するコンパレータ



〈図20〉 レベル・シフト効果を作る回路



ー2.1V以下にならないようにすることです。

口絵の写真16にレベル・シフトの一例を示します。

### ● モザイク効果

モザイク効果は見る機会も多いので、どのような効果かは説明する必要もないかもしれませんが、口絵の写真17にその例を示します。

原理は、画面を縦、横いくつかのマス目状に分けて、例えば、縦10画素(ライン)、横10画素を一つのマス目として画面を考え、そのマス目の中のデータは全部同じであるとするれば良いのです。

画像処理装置などでこれを行うと、一度全画面の画像データを記憶しておいて、上記の処理をソフト的に行います。エフェクタでは、あくまでリアルタイムに映像を変化させなければなりませんから、これをハード的に行わなければなりません。

モザイクの例として次に二つの方法を示しますが、前者はこれまでの回路を使ってモザイクの効果を出すもの、後者はメモリを使ってやや本格的なモザイク効果を作ったものです。

### ▶ モザイク(1)

まず、メモリを使わずにモザイク的效果を出す方法ですが、これには今までの回路、例えばディフェクト効果の回路で、4ビットをフルに出している状態で行

す。

また、 $V_{RT}$ と $V_{RB}$ 入力のインピーダンスも、TDC1021はほかのものに比べて高インピーダンスですから、OPアンプの出力をそのまま入れることができます。TDC1021以外のA-Dコンバータを使うときは、低インピーダンスのものもあるので注意してください。

図20で、ゲート信号には前述のワイプやスリットで使ったゲート信号をTTLレベルで入力してください。

調整は、まず $V_{RS}$ を最小にしておき、 $V_{RV}$ で $V_{RT}$ を設定します。オシロスコープで見ながら、黒レベルに設定します。出力される映像をモニタ・テレビでモニタしながら $V_{RS}$ を上げていくと、ゲートのかかったところだけコントラストが変化します。

ここで注意しなければならないのは、 $V_{RT}$ の電圧が $V_{RB}$ 以下にならないこと、また、最大定格に示される

絶賛発売中!



CQ出版社

ハードウェア・デザイン・シリーズ

★電子回路部品活用ハンドブックにつづく第2弾!

# アナログIC 活用ハンドブック

2色刷

B5判 320頁  
定価 1,850円

内容

- (1) プロローグ
- (2) OPアンプの基本機能
- (3) OPアンプと線形回路
- (4) 非線形の演算回路
- (5) フィルタ回路
- (6) 発振回路およびV-Fコンバータ
- (7) D-Aコンバータ
- (8) A-Dコンバータ
- (9) 電源用IC
- (10) スイッチング・レギュレータ



います。

A-D変換のサンプリング・クロックは、1ライン256画素程度に設定しましたが、これを128, 64, 32……と下げていきます。口絵の写真18にその例を示します。

すると、水平方向の何画素かが同じデータになるため、縦にシマ状にモザイク的な画像になります。

これは、A-Dコンバータがフラッシュ型なので、出力データはクロックの立ち上がりにはサンプリングされた値が、次のクロックまで保持されているためです。この回路を図21に示します。

A-Dコンバータのサンプリング・クロックは、水平アドレスからとります。水平アドレスはそのまま、1, 2, 4, 8, ……128画素ごとのクロックになります。

この方法では、水平方向は画素を大きく(粗く)することができるのですが、垂直方向、つまりライン方向は、1ラインごとに値が変わってしまいます。

しかし、場合によってはこの効果のほうが面白いこともあるので試してみてください。

#### ▶モザイク(2)

モザイク(1)では1ラインごとにしか画素を粗くすることができませんでした。これはいわばA-Dコンバータのラッチ機能をメモリ代わりに使っていたためです。次のラインまでデータを保持させるには、やはりメモリを使わなければなりません。

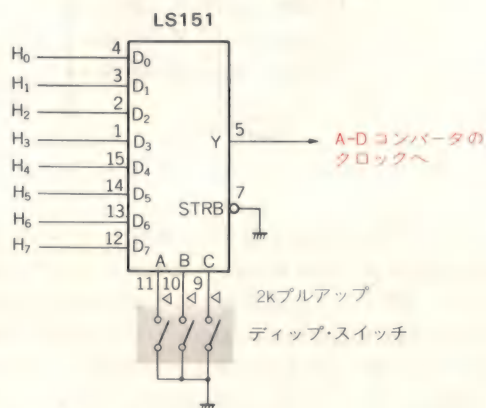
メモリはスタティックRAMを使います。スピードは100ns以上必要です。ここでは4ビット構成で、1024ワードとれるμPD2149Dを使いました。

ワード数は、1ラインの画像データを最大256と考えていますから、これで十分です。

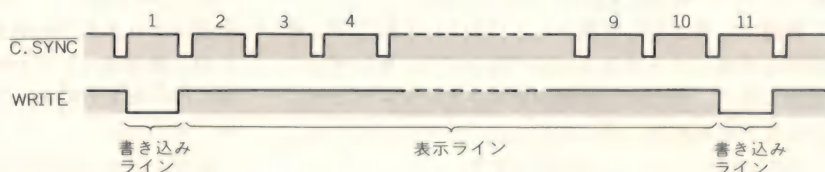
このモザイクの原理は、今モザイクの一つのブロックを10画素×10画素と考えると、最初のラインで、10画素ごとにA-D変換のサンプリングを行い、そのデータをメモリに記憶しておきます。このとき、メモリの水平側アドレスは256とれるようにします。

そして、図22に示すように、その次のラインからは

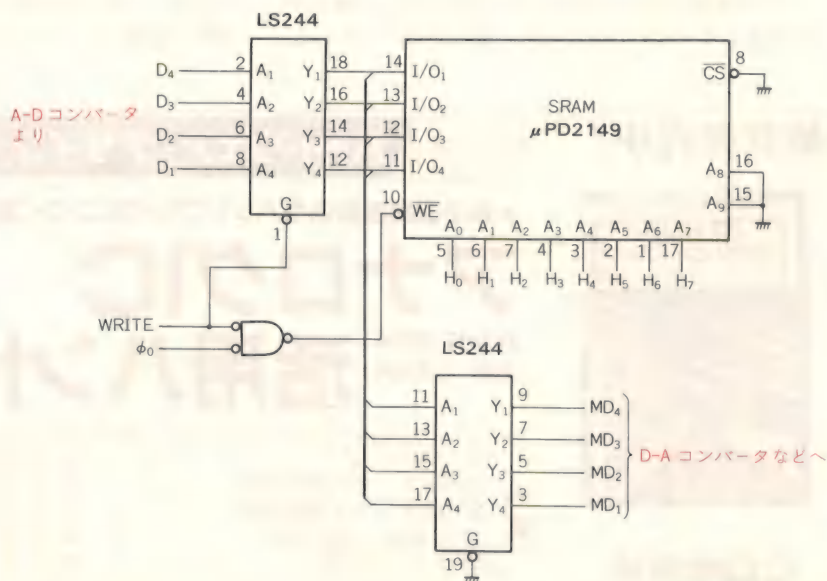
〈図21〉モザイク効果(1)を得る回路



〈図22〉  
ライン・サンプリングの  
タイミング

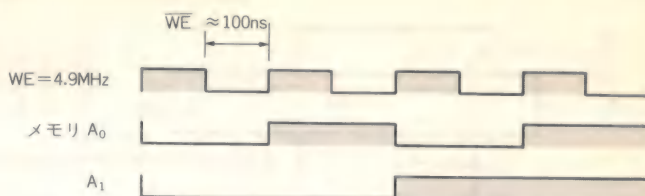


〈図23〉  
モザイク(2)のための  
メモリ回路





〈図24〉メモリへの書き込みタイミング



サンプリングを行わずに、1ライン目のサンプリング・データをメモリから呼び出して表示させます。そして10ライン目まで表示し、その次のラインでまたサンプリングを行いメモリに書き込みます。これを繰り返して256ラインの表示を行います。

回路は図23に示すように、1個のメモリと2個のバッファから成っています。A-D変換のデータはバッファを通してメモリに入っていますが、**バッファのゲートに、書き込み期間だけバッファがONするようにゲート信号を入れてやります。**

また、**メモリのデータ出力には、もう一つバッファを接続して、D-Aコンバータへ信号を送り込みます。**こちらのほうはゲートはONしっぱなしで、書き込み中はA-D変換の出力データを、読み出し中はメモリの出力データを出力します。

また、メモリのアドレスには水平アドレス8ビットを入れて、1ライン256画素に対応させています。

書き込みは、図24に示すように、約100nsの書き込みパルスによってメモリへ書き込みます。

モザイクの大きさを $n$ 画素とすると、書き込みは $n$ ラインごとに、最初の1ラインだけ書き込みを行います。図25に示す回路によって書き込みのタイミングを与えます。

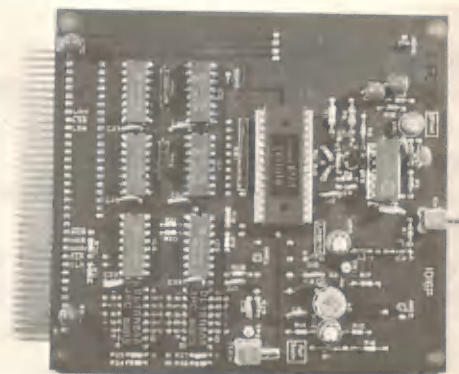
この回路では、**モザイクの粗さはディップ・スイッチによって設定**するようになっています。これはバイナリで入れなければならないので、ロータリ式の4ビット・ディップ・スイッチを2個使うと便利かもしれません。あるいは、ディップ・スイッチのところにアップ/ダウン・カウンタの出力を入れて、スイッチによって画素を粗くしたり細かくしたり変化させるとさら

## 実験用ビデオA-Dコンバータ

これから画像処理回路について勉強したいという場合、デジタル技術との融合という意味で、ビデオ用A-Dコンバータとのつき合いは避けることができません。しかし、アナログ回路の知識が少ない人が、**ビデオ帯域（～20MHz）のアナログ信号を扱うことは、そう容易ではありません。**

ビデオA-DコンバータのICが国産化され、低価格化されてきたのは事実ですが、使い方がやさしくなってきたわけではありません。使ううえでの技術としては、相変わらず10万円以上していた時代と同等の高度なアナログ技術（特に実装を主とした技術）が必要です。

そこで、はじめてビデオA-Dコンバータを扱うという場合、あるいは実験的に扱うという場合には、各ICメーカーで用意されている評価用ボードの使用をお勧めします。写真Aにその実例を示しますが、これらのボードには75Ω同軸コネクタによるアナログ入力、入力バッファ・アンプ、基準電圧、**ビデオA-Dコンバータ、ECL⇄TTLインターフェース回路などが装備**されており、電源とクロックさえ供



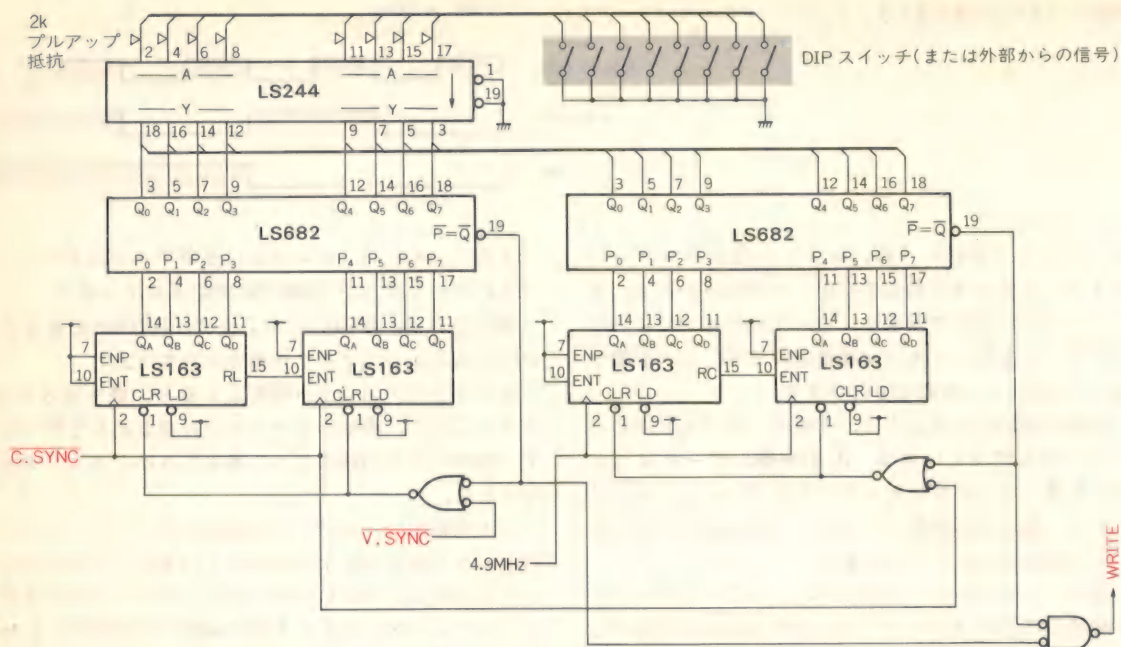
〈写真A〉CXA1056P 50MHz A-D変換ボード  
〔ソニー(株)〕

給すれば、すぐに実用できるように構成されています。

これらのボードは、見るとよくわかるのですが、**ベタ・アースを基本にしてプリント基板パターンが設計**されており、電源パスコンのレイアウトなども含めて、製品設計の際のサンプルとしても大いに役立つものです。  
(編)



〈図25〉 書き込みタイミングを作る回路



に効果的です。

これまでに説明した各エフェクトではカラー信号の処理をしていませんでしたが、カラー化をするとなんとなくデコーダが必要になります。また、メモリも3個、RGB用に必要になります。当然A-DコンバータやD-Aコンバータ、ビデオ・アンプも3倍必要になります。

そのため、今回は特にカラー対応を完全に行っているわけではありませんので、特にモザイクでは画素とカラー信号の1対1対応はとれていません。モニター・テレビで見る限り、なんとかカラー画像としてモニターすることができますが、ややにじんだ感じになるかもしれません。

また、NTSC信号として厳密に扱っているわけではないので、個人的な使い方をする以外の場合は、専門

書を参照してください。

### ● エフェクタの使い方

以上で回路の説明は終わりです。特にむずかしい所もないので容易にビデオ・エフェクトの映像を見ることができると思います。

ディフェクトのほうは、カラー・カメラまたはVTRのカラー・ビデオ信号を入力してください。コントラストのはっきりした映像のほうが効果的です。

効果をかけた映像はVTRに録画できますから、自分で編集する作品の一部に作ってみるとよいでしょう。

疑似カラーのほうは、むしろ白黒カメラのほうが効果的です。人物をとると少し異様な感じになってしまいますが、部屋の中の小物をとってみるとビデオ・アートのように非常に面白いと思います。



ビデオ・フィールド・メモリの設計・製作



岩崎 潔

デジタル・スチルとも呼ばれる、テレビ1画面分をデジタル的に記憶させるメモリについて紹介します。専用のシリアル・アクセス・メモリの使用により設計が容易になっています。

画像処理技術を大きく推進させた原動力は、なんといっても大容量メモリと高速A-Dコンバータの普及でしょう。これらが容易に利用できるようになったことで、画像処理のデジタル化が可能になり、身近な技術でも画像が取り扱えるようになってきました。

そこで、この章ではNTSCコンポジット信号をA-D変換してデジタル化し、テレビの1画面分を画像メモリへ取り込んでフリーズ(スチル化；静止画)するフィールド・メモリの設計法について紹介します。

画像専用シリアル・アクセス・メモリ

● 1画面分を記憶するために必要なメモリの性能  
テレビの1画面分をデジタル的に記憶・保持させるフィールド・メモリの構成を図1に示します。NTSCコンポジットのアナログ信号はいったんデジタル信号に変換してから記憶しますが、A-D変換用のサンプリング周波数は、理論的には映像信号の最高周波数の2倍以上の周波数でなければなりません(標準化定理)。

そこで、コンポジット信号(NTSC信号)の変換を行うには、通常は色副搬送波周波数の3倍、または4倍の周波数でサンプリングします。サンプリングされたデータは、A-D変換してデジタル信号としてメモリに書き込みます。その後、必要とするタイミングで読み出し、D-A変換することにより、アナログ信号に復元するわけです。

ここでメモリ側に要求される性能を考えてみます。カラー・テレビ信号としてNTSC信号を例にとると、色副搬送波周波数 $f_{sc}$ は3.579545MHzです。したがって、サンプリング・レートをその3倍とする場合、メモリ側の処理スピード(サイクル時間)は、

$1 / (3579545 \times 3) \approx 93.1(\text{ns})$   
となります。

また、映像信号1フィールド(1画面)分の記憶を考える場合に必要とされるメモリの容量は、量子化ビット数を8ビットとすると、

$$\frac{525}{2} \times \frac{455}{2} \times 3 \times 8 = 1,433,250 \text{ (ビット)}$$
  
となります。

同様に、サンプリング・レートを色副搬送波周波数の4倍とする場合についても計算を行った結果を表1に示します。

このように、画像メモリは一般に大容量かつ、93.1(ns)や69.8(ns)もの高速のメモリ・サイクル時間を同時に満足させなければなりません。

高速のバイポーラRAMを使用すると、処理スピードについては十分です。しかし、大容量化には不経済であるばかりでなく、消費電力も大きく熱的な問題も起こりかねません。したがって、大容量で低電力の既存の汎用DRAMを使用することになるのですが、高速のメモリ・サイクル時間を実現するのは簡単ではありません。

そこで、メモリを複数個同時アクセスし、その並列データを直列データに変換するパラレル-シリアル変換などの処理を行い、見かけ上のサイクル時間を速くすることが考えられます。しかし、これには応用上で複雑なタイミングを作ってやらなければなりません。

〈表1〉画像メモリに要求される処理スピードと容量

サンプリング・レート	メモリ・サイクル時間	容量(ビット数)
3 $f_{sc}$	93.1 ns	1,433,250ビット
4 $f_{sc}$	69.8 ns	1,911,000ビット

(注)  $f_{sc}$ は色副搬送波周波数、3 $f_{sc}$ は $f_{sc}$ の3倍  
4 $f_{sc}$ は $f_{sc}$ の4倍

〈図1〉画像の記憶(映像信号の流れ)





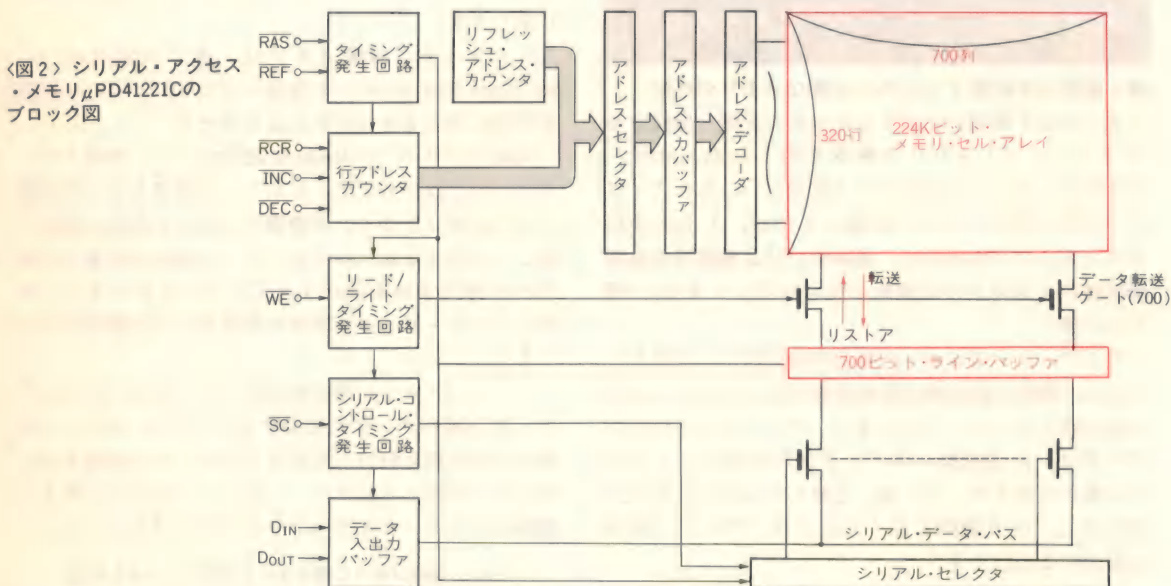
## ● シリアル・アクセス・メモリ $\mu$ PD41221C

テレビやVTRなどのシステムでは、走査線変換(ランダム・スキャン、ラスタ・スキャン)や、図形処理分野の種々の複雑な処理を行わないものがほとんどです。つまり、画像用としてのメモリのアクセスは複雑なものではなく、シリアル・アクセスがもっとも有効な方法です。

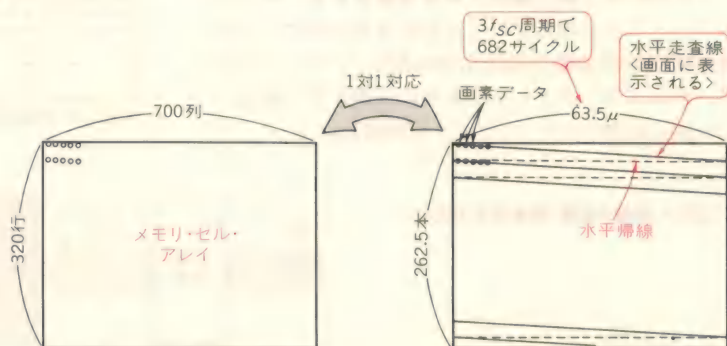
図2に、画像メモリ用として作られたシリアル・アクセス・メモリ $\mu$ PD41221Cのブロック図を示します。このICは、デジタル符号化された映像信号を、あるサンプリング・レートでシリアルに書き込むことにより、容易に1フィールド分の映像信号の記憶を可能としています。特徴を次に示します。

### (1) 224,000ワード×1ビット構成のメモリ・セル

メモリ・セルの構成を224,000×1ビット(320行×700列)とし、1フィールド分の映像信号のイメージと、1対1に対応させてあります。これは図3に示すように、テレビの水平走査線に各行を、またデジタル化されたデータを水平走査線上の画素データにそれぞれ対応させてあります。



〈図3〉 メモリ・セルと映像信号の対応



### (2) 連続700ビットの高速シリアル・アクセス

700ビットのライン・バッファを内蔵することにより、連続700ビットまでの高速シリアル・アクセスができます。つまり、1水平走査線期間分のデータのシリアル・アクセスを可能としています。

### (3) 非同期リフレッシュ

256K DRAMの標準プロセスで設計されたメモリ・セルは、1トランジスタ・ダイナミック型(コンデンサへの電荷の充放電により記憶)を用いていますので、リフレッシュが必要となります。しかし、このICはライン・バッファへデータを転送した後は、メモリ自体の動作はシリアル・アクセス動作と特に関係なくなるため、シリアル・アクセス中に非同期にリフレッシュ動作が可能となっています。

また、アドレスの設定については、アドレス・カウンタを内蔵していますので、 $\overline{\text{INC}}$ 、 $\overline{\text{DEC}}$ 、 $\overline{\text{RCR}}$ 入力により、システム構成が容易となっています。

ICの出力特性に関しては、

$$V_{OH}/V_{OL}=0.7V_{CC}/0.2V_{CC}$$

となっており、HC-MOS負荷に対応しています。



## ● シリアル・アクセス・メモリの動作

μPD41221Cに一画面分のデータを書き込むには、**シリアル・ライト→データ・リストア**と呼ぶサイクルを実行します。また、メモリよりデータを読み出す際には、**データ転送→シリアル・リード**と呼ぶサイクルを実行します。この動作を水平走査線ごとに行えば、一画面分の映像信号データの記憶・読み出しが可能となります。

### ▶ シリアル・ライト/リード・サイクル

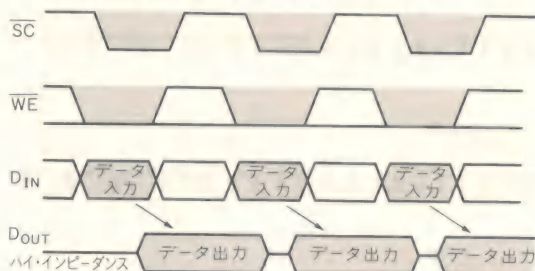
シリアル・ライト・サイクルは、 $\overline{SC}$ クロック入力時に $\overline{WE}$ を“L”にすることで実行され、 $\overline{SC}$ クロックの立ち下がりエッジでデータが取り込まれます。また、この時に入力されたデータは $D_{OUT}$ に同一アクセス時間で出力されます。そのため、ライト・サイクル中にもこの出力を直接用いることが可能で、データの入出力系にパイパスをもたせる必要はありません。

シリアル・リード・サイクルは同様に、 $\overline{WE}$ を“H”とする時に、 $\overline{SC}$ クロックの立ち下がりエッジでシリアル・データが読み出されます。シリアル・ライト・サイクル/リード・サイクルのタイムチャートを図4、図5に示します。

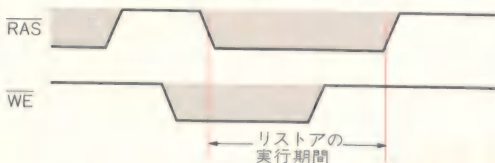
なお各タイミングにおいて $D_{OUT}$ は一度ハイ・インピーダンスとなる期間がありますが、負荷としてHC-MOSを考慮する場合は、C-MOSの入力インピーダンスが極めて高いため、実際にはデータ出力は次のデータ出力まで保持されます。

### ▶ データ・リストア/転送サイクル

〈図4〉 シリアル・ライト・サイクル



〈図6〉 データ・リストア・サイクル



〈図8〉 行アドレス・カウンタ・サイクル



高速シリアル・アクセスのためにライン・バッファに蓄えるデータは、メモリ・セル・アレイとの間で、入出力を行う必要があります。したがって、700ビットのデータ転送ゲートを経由させて、 $\overline{RAS}$ 、 $\overline{WE}$ のタイミングにより、ライン・バッファとメモリ・セルの**1行700ビットのデータ**について、**リード/ライト動作**を実行します。

$\overline{RAS}$ クロックの立ち下がり時に、 $\overline{WE}$ を“L”とすることにより、図6のようにデータ・リストア・サイクル(ライン・バッファのデータがメモリ・セルに書き込まれる)を実行します。また、 $\overline{WE}$ を“H”とすることにより、図7のようにデータ転送サイクル(メモリ・セルのデータがライン・バッファに読み出される)を実行します。

### ▶ 行カウンタ・コントロール・サイクル

3本の行アドレス・コントロール・クロックは、それぞれ次のように行アドレスをコントロールします。

$\overline{INC}$ : 行アドレスを+1

$\overline{DEC}$ : 行アドレスを-1

$\overline{RCR}$ : 行アドレスを0にリセットする

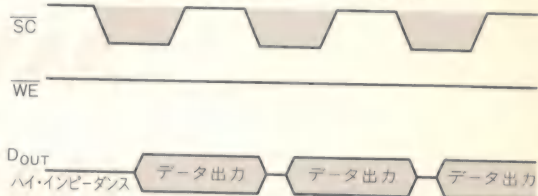
図8に行アドレス・カウンタのタイムチャートを示します。

フィールド・メモリのコントロールは、各水平走査線ごとに $\overline{INC}$ を1回ずつ、また $\overline{RCR}$ を各垂直走査期間に1回入力することで可能になります。図3で示したメモリ・セルを、映像信号に対応させるのが可能なことは、容易に理解できるでしょう。

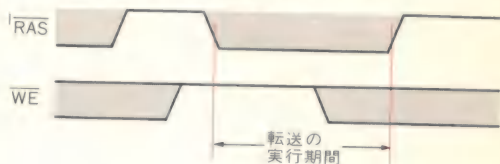
### ▶ リフレッシュ・サイクル

リフレッシュは図9に示すように、基本的に $\overline{RAS}$

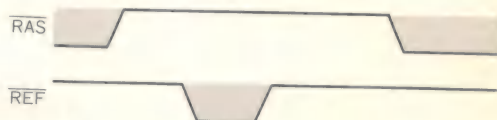
〈図5〉 シリアル・リード・サイクル



〈図7〉 データ転送サイクル



〈図9〉 リフレッシュ・サイクル





クロックが“H”の期間に、シリアル・サイクルとは非同期にREF信号を入力することにより、1行分のリフレッシュが実行されます。

このICのリフレッシュは、2 ms以内に320回以上のクロックを入力することで実行できますので、フィールド・メモリを構成するときには、

$$2 \text{ ms} / 320 = 6.25 \mu\text{s}$$

$$> 63.5 \mu\text{s} / 11$$

から、1 水平走査期間に11回以上のリフレッシュ・クロックを入力します。

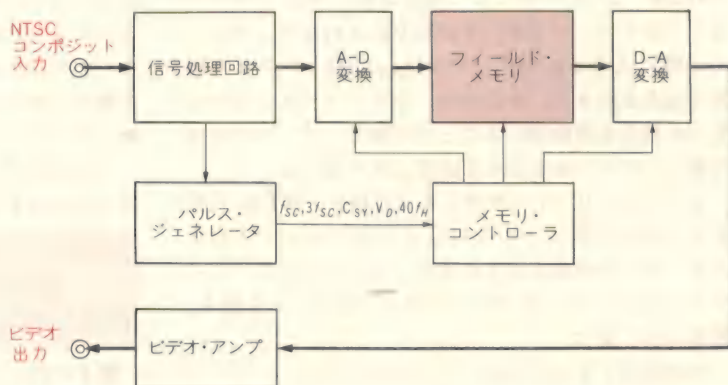
## フィールド・メモリの構成法

図10にNTSCコンポジット信号のフィールド・メモリ・システムのブロック図を示します。以下、各部の設計法を紹介します。

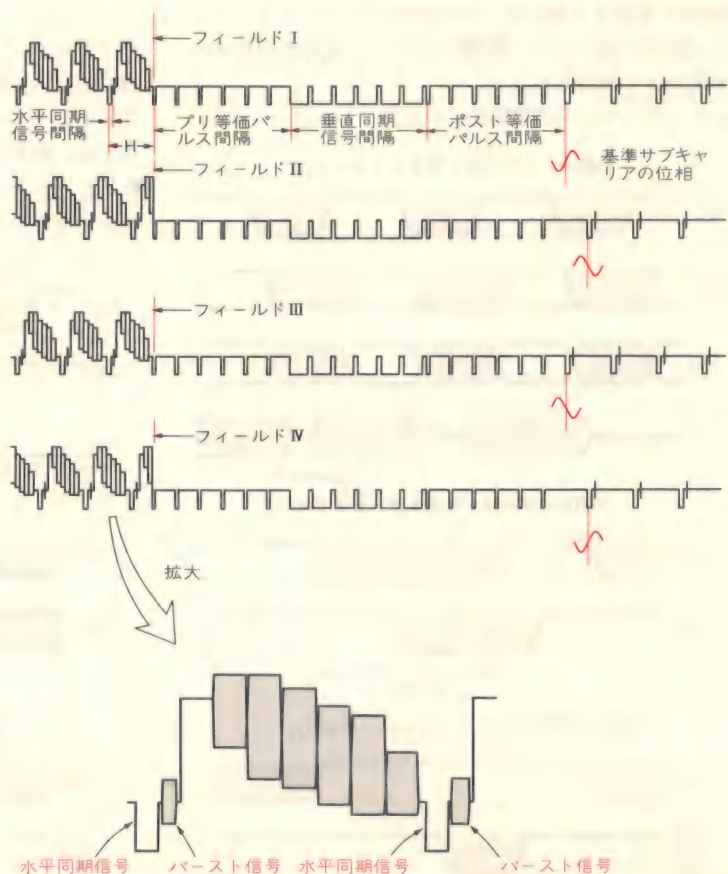
### ● 信号処理、パルス・ジェネレータの考え方

ここでは、入力されるコンポジット信号から、システムを構成するうえで基準となるクロックを発生します。またコンポジット信号そのものをA-D変換に適

〈図10〉  
フィールド・メモリ・システムのブロック図



〈図11〉 NTSCコンポジット





した信号に変換します。

このためには図11のコンポジット信号からパースト信号を分離します。そして図12に示すように、パースト信号に同期したクロック $f_{sc}$ (=3.58MHz)と、これを3てい倍した基準クロック $3f_{sc}$ を作ります。

また、同期信号分離回路により複合同期信号 $C_{sy}$ を分離します。そして、この $C_{sy}$ から垂直同期信号 $V_D$ 、および水平同期信号の40てい倍信号 $40f_H$ を作り、 $f_{sc}$ と $3f_{sc}$ と共にメモリ・コントローラに送ります。

これらの信号を作るのは大変そうですが、VTRのカラー信号処理用に作られた $\mu PC1536C$ を利用すると、比較的簡単に構成することができます。このICの概要は第6章の図15(57ページ)を見てください。

さらに入力ビデオ信号については、同期信号の先端を一定電位に固定するシンク・チップ・クランプ回路を通して、A-D変換回路に供給します。

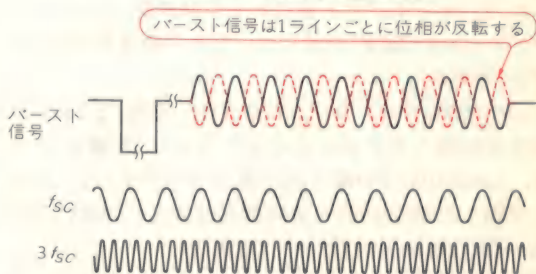
この回路もVTR用輝度信号処理ICの一部を使います。型名は $\mu PC2310C$ で、図13にこの構成を示します。

## ● 信号処理、パルス・ジェネレータの回路構成

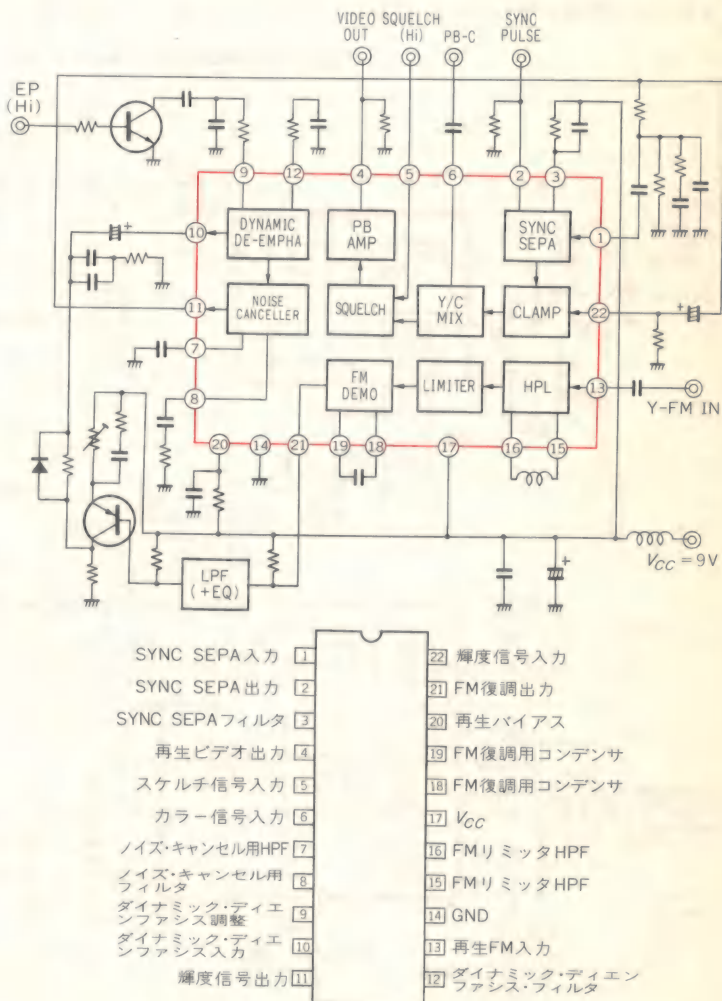
図14に信号処理、パルス・ジェネレータ部の回路図を示します。ビデオ信号は、 $\mu PC2310C$ の22番ピンに入力されます(写真1)。そして、シンク・チップ・クランプを受け、4番ピンに $2V_{P-P}$ で出力されます(写真2)。

シンク・チップ・クランプは、同期信号の先端電圧を一定電位に固定する回路のことで、入力信号の各ラインのDCレベルを一定にし、D-A変換によるグリッ

〈図12〉 パースト信号と $f_{sc}$ ,  $3f_{sc}$

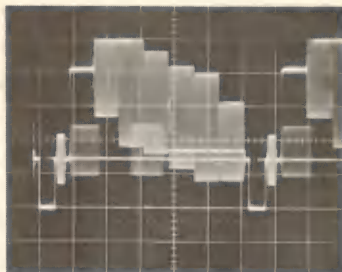


〈図13〉  
VTR用輝度信号処理IC  $\mu PC2310$ の構成





〈写真1〉  
入力コンポジット信号  
[0.2V/div, 10 $\mu$ s/div]



〈写真2〉  
コンポジット信号と  
VR<sub>1</sub>の調整  
[1V/div, 10 $\mu$ s/div]



チなどが画面に対して与える影響を軽減しています。

一方、コンポジット信号から3.58MHzのバンドパス・フィルタ(BPF)を通して、カラー信号を抜きとります(写真3)。

$\mu$ PC1536Cでは、まず、このカラー信号のバースト信号と同期したクロック $f_{sc}$ を作ります(写真4)。また、 $\mu$ PC2310Cで分離された複合同期信号より、これと同期した $40f_H$ 信号(水平同期周波数 $f_H$ の40倍の周波数 $\approx 629.37$ kHz)を作ります(写真5)。

また $f_{sc}$ 信号は、HC-MOSのインバータを使用して、きれいな方形波に整形します(写真6)。さらに、これ

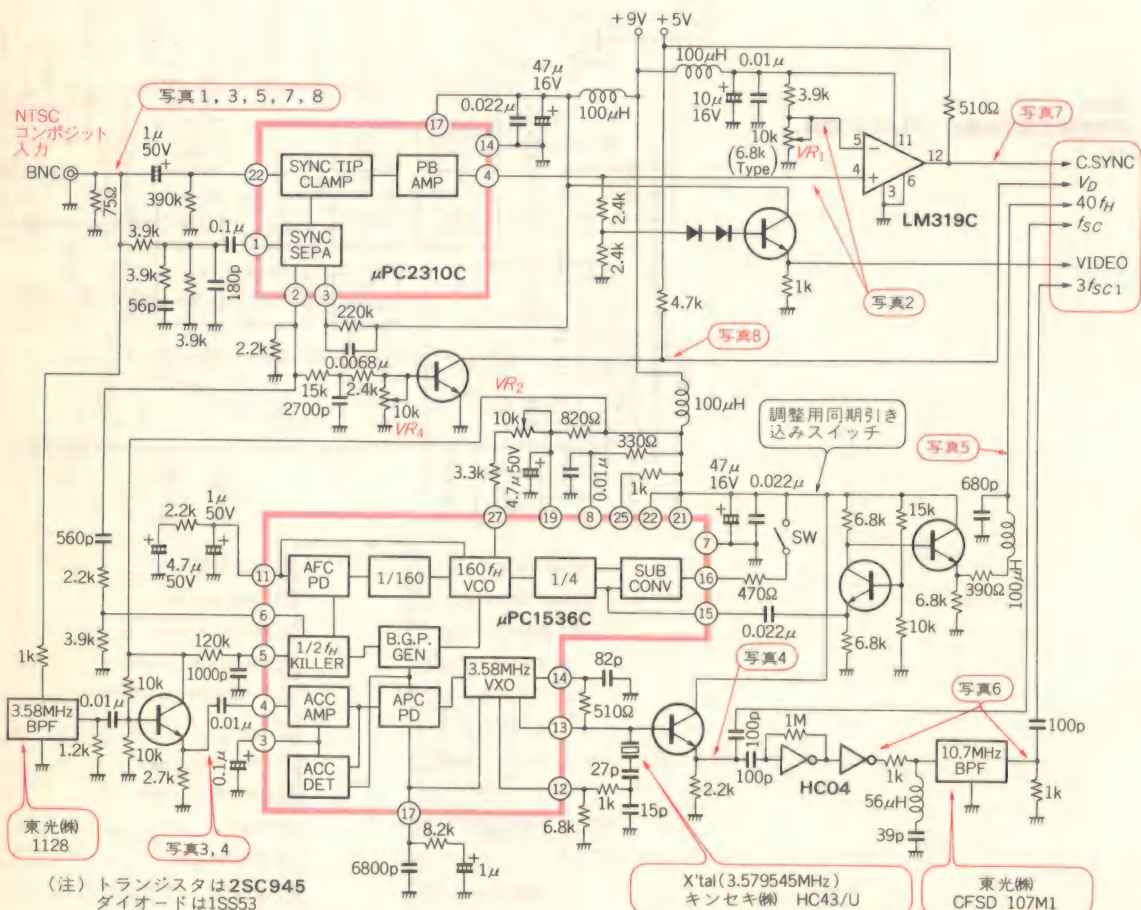
から3倍の高調波 $3f_{sc}$ ( $\approx 10.7$ MHz)を抜き取り、A-D/D-A変換、メモリ・コントロールの基準クロックに使用します。

メモリのコントロールは高速に行う必要がありますが、 $\mu$ PC2310Cの同期分離回路では遅延時間が約300nsありますので、高速のコンパレータLM319C(遅延時間 $\approx 80$ ns)を用いて、同期信号を分離しています(写真7)。

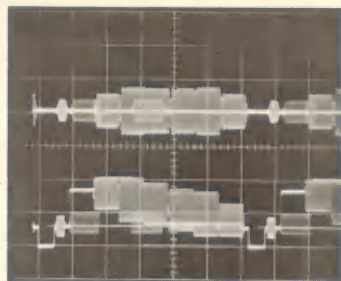
### ● メモリ・コントローラ、A-D/D-A変換、フィールド・メモリの考え方

メモリ・コントローラでは、パルス・ジェネレータ

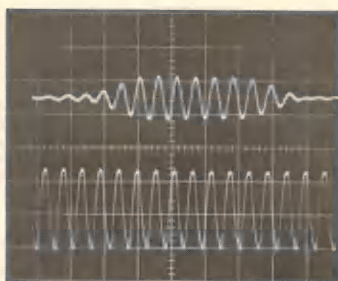
〈図14〉 信号処理回路のパルス・ジェネレータ



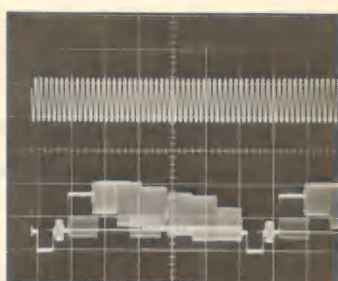




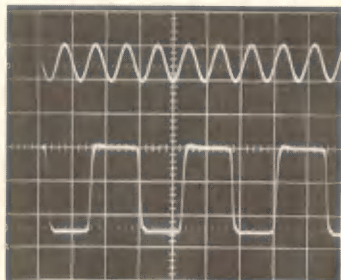
〈写真3〉入力コンポジット信号(下)と  
カラー信号(上)  
〔上: 0.2V/div, 下: 0.5V/div, 10 $\mu$ s/div〕



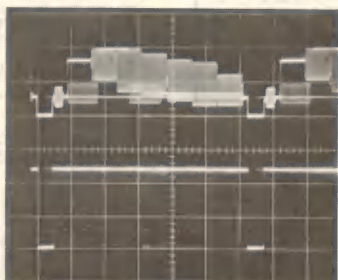
〈写真4〉バースト信号(上)とカラー・  
サブキャリア・クロック $f_{sc}$ (下)  
〔上: 0.1V/div, 下: 0.2V/div, 0.5 $\mu$ s/div〕



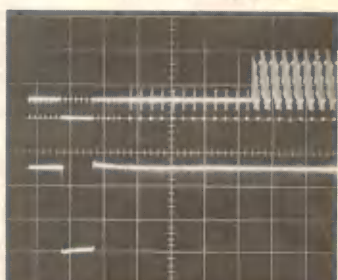
〈写真5〉入力コンポジット(下)と $40f_H$   
信号(上)  
〔0.5V/div, 10 $\mu$ s/div〕



〈写真6〉 $f_{sc}$ の方形波(下)とメイン  
クロック $3f_{sc}$ (上)  
〔上: 0.5V/div, 下: 2V/div, 0.1 $\mu$ s/div〕



〈写真7〉入力コンポジット信号(上)  
と同期信号(下)  
〔上: 0.5V/div, 下: 0.2V/div, 10 $\mu$ s/div〕



〈写真8〉入力コンポジット信号(上)  
と垂直同期信号 $V_D$ パルス(下)  
〔上: 0.5V/div, 下: 2V/div, 0.2ms/div〕

より供給される $f_{sc}$ ,  $3f_{sc}$ ,  $C_{SY}$ ,  $V_D$ ,  $40f_H$ 信号から,  
 $\mu$ PD41221Cを制御する信号( $\overline{RAS}$ ,  $\overline{WE}$ ,  $\overline{SC}$ ,  $\overline{REF}$ ,  
 $\overline{INC}$ ,  $\overline{RCR}$ )を発生します。

A-D/D-A変換のサンプリング・クロックは $3f_{sc}$ (=  
10.7MHz)を用います。これはまた、メモリ・コント  
ロールの基準となるクロックにもなっています。量子

化ビット数に関しては任意に設定できますが、通常は  
6あるいは8ビットが用いられます。フィールド・メモ  
リ $\mu$ PD41221Cは1ビット分の信号しか扱うことが  
できませんので、量子化ビット数に合わせた数量が必要で

図15にA-D/D-Aコンバータ、およびメモリの構成

## 1Mビット映像メモリ

メモリICの発展、ことに容量や速度の面での進  
歩には目を見張るものがあります。コンピュータ用  
のメモリとしては、すでに1MビットDRAMが主  
流になりつつありますが、これからの市場として有  
望視されている映像メモリの分野にも、同様に1M  
ビットのものが登場しています。

本文でも紹介があるように、NTSC方式のテレ  
ビ信号の1フィールド(1/60秒に送られてくるテレ  
ビの画像情報)を記憶するためには約1.8Mビット  
、1フレーム(1フィールドの2枚分のテレビ画  
像情報)を記憶するためには、約3.7Mビットの容量  
が必要で、さらに画像をリアルタイムに取り込むた  
めにはサイクル時間が約70ns以内でなければならないと  
されています(サンプリング周波数 $4f_{sc}$ 、量子  
化が8ビットの場合)。したがって、この分野で

もメモリの大容量化と高速化とが欠かせないのです。

最近サンプルが用意されだした**1Mビット映像メモ**  
**リ**について主なものを紹介しておきましょう。(編)

### ▶ HM53051P〔㈱日立〕

構成: 256K $\times$ 4ビット, 60ns, 18ピンDIP,  
アドレス発生回路内蔵。

### ▶ TMS4C1050〔日本T I ㈱〕

構成: 256K $\times$ 4ビット, 60ns/30ns, 16ピンDIP,  
非同期シリアル書き込み/読み出し方式。

### ▶ MN4700〔松下電子工業㈱〕

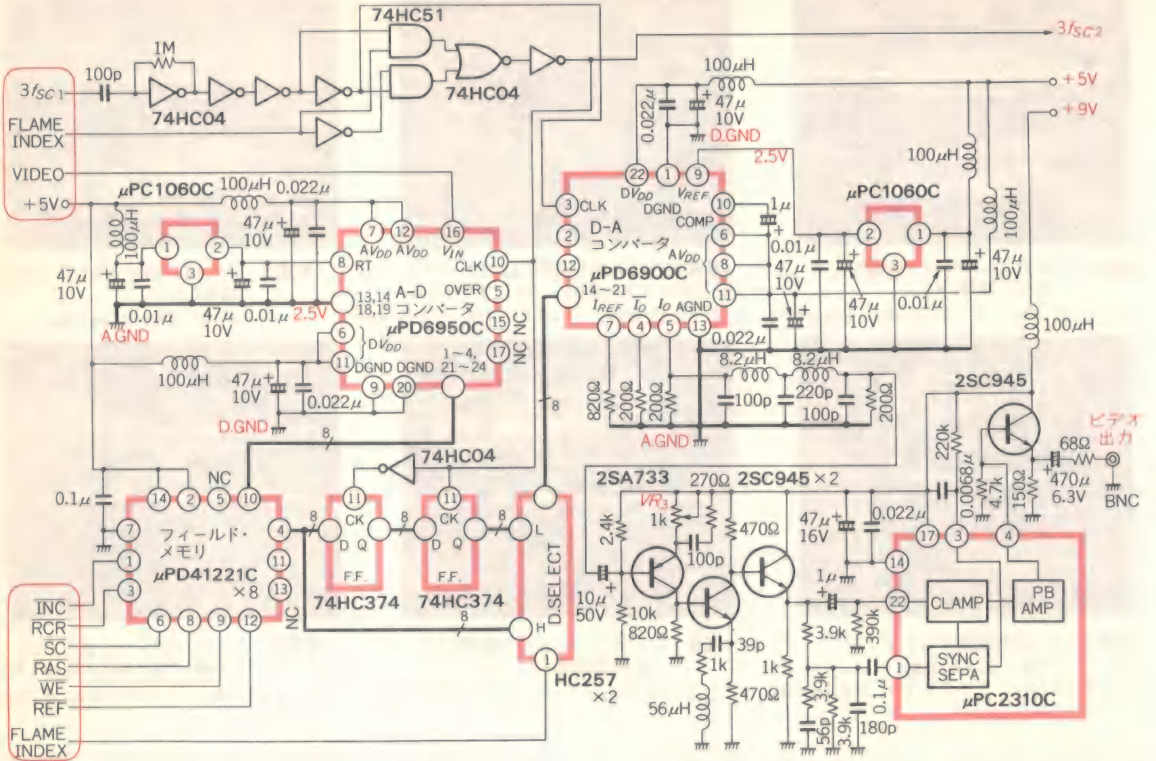
構成: 256K(32K $\times$ 8ビット・シリアル) $\times$ 4ビット,  
30ns, 40ピンDIP, シリアル入出力。

### ▶ TC521000P〔㈱東芝〕

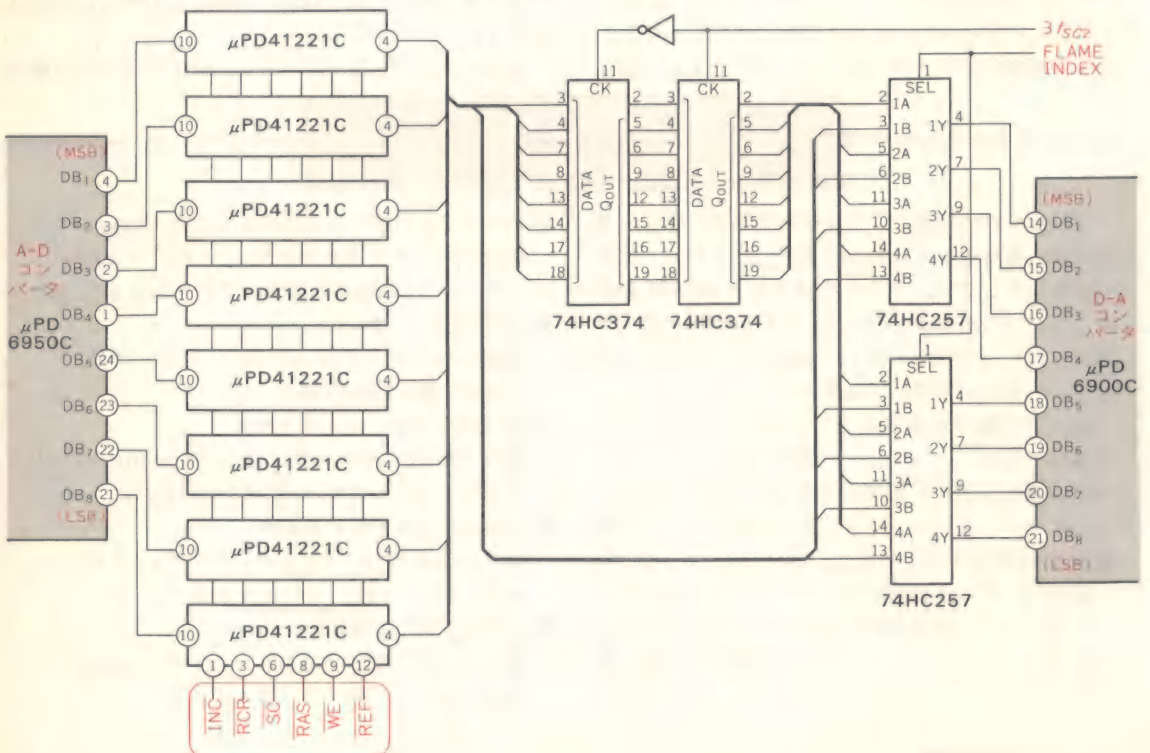
構成: 256K(32K $\times$ 8ビット・シリアル) $\times$ 4ビット,  
30ns, 40ピンDIP, シリアル入出力。



〈図15(a)〉 A-D/D-A変換、フィールド・メモリ(1)



〈図15(b)〉 A-D/D-A変換、フィールド・メモリ(2)





について示しますが、ここでは8ビットの分解能を期待することにして、A-Dコンバータに $\mu$ PD6950C、D-Aコンバータに $\mu$ PD6900Cを使うことにしました。したがって、使用するメモリ $\mu$ PD41221Cは8個となります。なお、A-D/D-Aコンバータのそれぞれには基準電圧が必要ですが、ここでは、2.5V出力の基準電圧IC  $\mu$ PC1060Cを使用しています。

さて、A-D/D-Aコンバータの詳細な動作説明についてはここでは省略しますが、その周辺設計についてはグラウンドまわりに特に注意が必要です。ピン接続については69ページに示してあります。

図15に示しているDGND(ディジタル系グラウンド)とAGND(アナログ系グラウンド)は分離し、できる限りシステムの主電源に近いところで接続するように設計します。

ディジタル系の信号は、アナログ系の信号にノイズとして直接影響を与えることが多く、グラウンドの引きまわし方でノイズ・レベルは変化するため、苦勞する部分が多くなります。

A-D変換用のサンプリング・クロックは $3f_{sc}$ ですので、理論上はこの1/2の周波数( $\approx 5.35\text{MHz}$ )までしか変換できません。これ以上の周波数を入力すると、折り返し誤差が出てしまいます。

実際のテレビジョン放送では、音声信号搬送波が映像信号搬送波よりも4.5MHz高い所にあります。したがって、映像信号に含まれる周波数成分は0から約4MHzまでに限られてしまいます。このシステムではA-D変換の前にローパス・フィルタを通してはいませんが、入力信号の周波数帯域が広がることが考えられる場合には、ローパス・フィルタにより、約5MHz以上の成分を十分に減衰させようとしてA-D変換するようにします。

D-A変換によりアナログ信号に復元したビデオ信号は、LC構成のローパス・フィルタにより、グリッチなどの変換ノイズを低減します。そして周波数特性補正を行った後、 $\mu$ PC2310Cの4番ピンから $2V_{P-P}$ で出力します。そして最後にインピーダンス変換により、75 $\Omega$ の出力インピーダンスでBNCコネクタより出力します。なおこの時にアナログのビデオ信号を処理する回路のグラウンドは、アナログ系グラウンドに接続することに注意します。

## ● メモリ・コントローラ

メモリ・コントローラの構成はやや複雑です。ブロックを分けて説明しましょう。図16が回路構成です。

### ▶ 水平同期信号 $H_D$ 発生回路

図17にタイムチャートを示します。同期信号には水平同期、垂直等化パルス、垂直同期の3種類の信号があります。図18に示すようなフィールド記憶を行うに当たり、各水平ラインにナンバを付ける必要があります。

す。

そのためには、同期信号は一定位相でなければなりませんので、垂直等化パルスや垂直同期信号に含まれる $(1/2)f_H$ のタイミングのパルスを消去するのがこの回路の役割です。

$FF_{21}$ のCKに入力される3種類のどの信号に対しても、 $FF_{22}$ のQに1H期間に1発のパルスを出力し、図19に示すように複合同期信号より水平同期信号 $H_D$ を得ることができます。

### ▶ フレーム・インデックス回路

メモリに記憶された信号を読み出す時には、フレームごとに何らかの処理を施す必要があります。そこで、各フレームにはインデックスを与えて区別しています。図20にこの部分のタイムチャートを示します。

カラー・サブキャリアの位相は、各ラインごとに反転しています。したがって、これを利用して各ラインに奇数ラインか偶数ラインなのかのインデックスを割り振ります( $FF_{26}Q$ )。そして垂直同期パルス $V_D$ と論理をとることにより、フレームのインデックスが可能となります( $FF_{27}\overline{Q}$ )。なお、 $V_D$ パルスの立ち上がりのタイミングは、垂直等化パルスの直後の垂直同期信号の期間にくるように、図14の積分回路を設計しています。

$FF_{23}$ の $\overline{Q}$ からは、各水平ラインごとにデューティが約50%の信号を出力しています。したがって、これと $V_D$ パルスの論理によりフィールド・インデックスが可能です( $FF_{24}Q$ )。フィールド・インデックスの立ち上がりエッジで、 $G_{48}$ より1発パルスを発生し、水平ラインのカウンタをフィールドごとによりセットしています。

なお、図20からも明らかなように、カウンタのリセット時は、実際のフィールドの始まりとは合致しません。カウンタの示す数値は、実際のライン $N_0$ よりも4少ない数値となりますので、この点は後の回路で考慮します。

### ▶ 行アドレス・コントロール・クロック発生回路

図21にタイムチャートを示します。 $CNT_5 \sim CNT_7$ により10ビットのバイナリ・カウンタを構成し、各水平ラインごとにカウンタ数値を順次進めます。

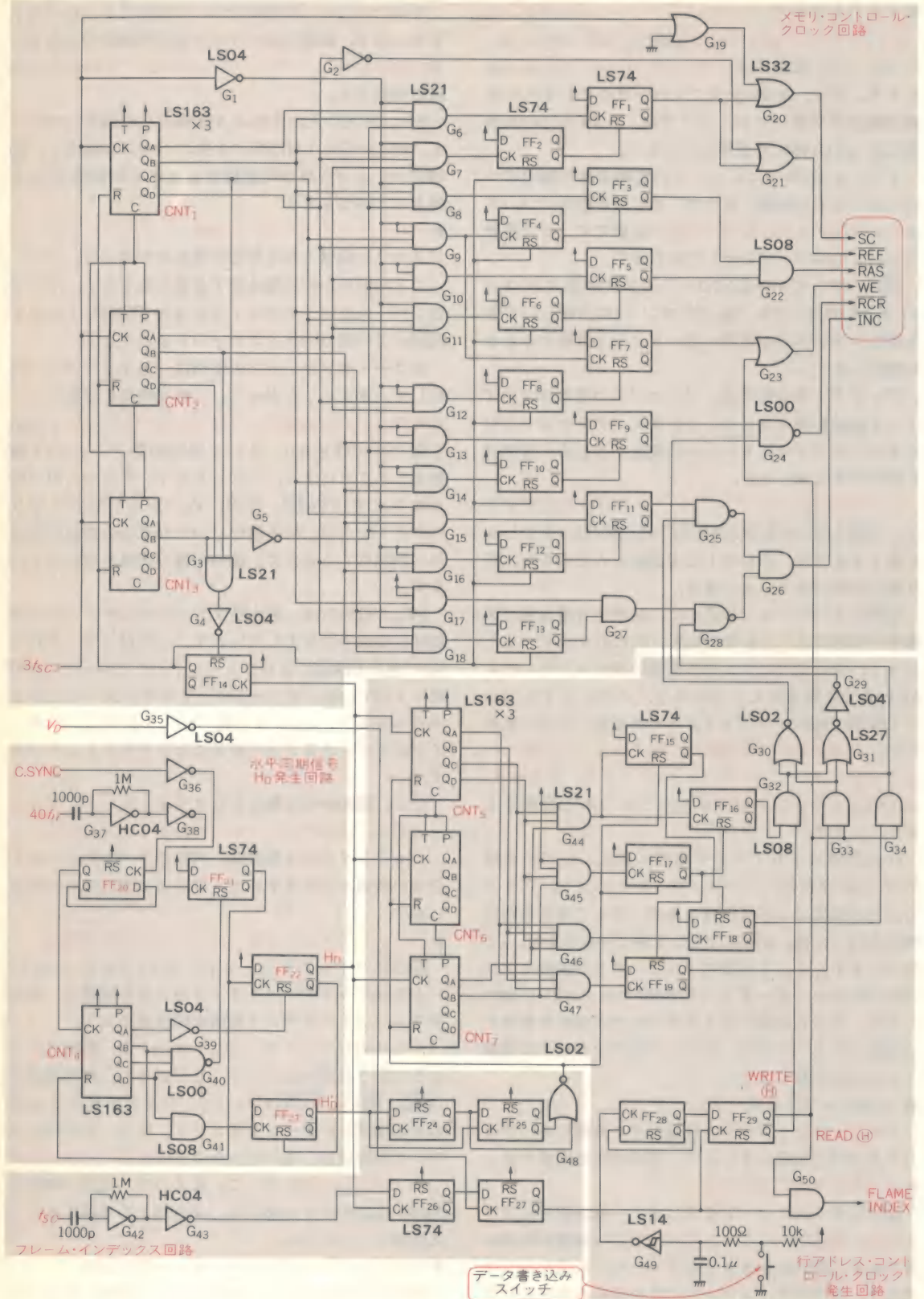
メモリの行アドレス・コントロールは、図22に示すように行う必要があります。したがって、各制御信号( $\overline{INC}$ ,  $\overline{INC}_2$ ,  $\overline{RCR}$ )について、パルスを与える必要のある期間だけゲートを開きます。また、この時のカウンタの数値は、先に述べたように実際のライン $N_0$ よりも4少ない数値になっていますので、図21の数値は図23や図20のライン $N_0$ では、4を加えて考慮することに注意してください。

### ▶ メモリ・コントロール・クロック発生回路

図24にタイムチャートを示します。基準となるクロ

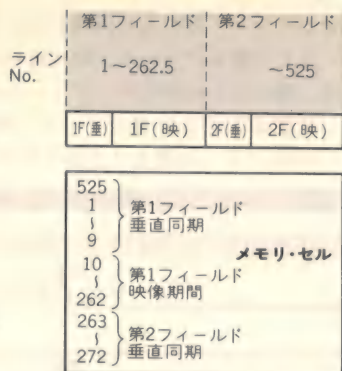


〈図16〉 メモリ・コントローラ





〈図18〉  
メモリ・セルの構成



ックとして、読み出し時のみフレームごとに位相反転する信号  $3f_{sc}$  を使い、フィールド・メモリ  $\mu PD41221$  Cを制御するクロックを発生します。

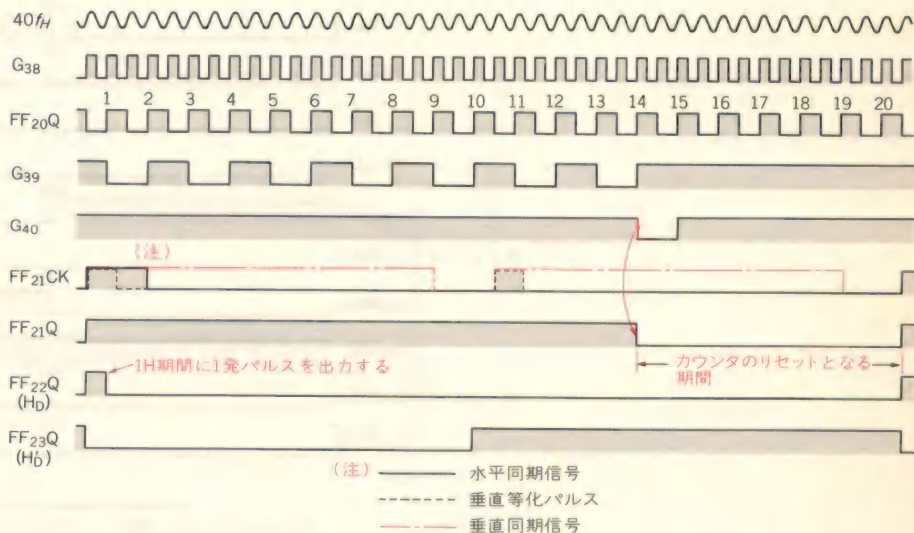
INC, INC<sub>2</sub>, RCRパルスについては常に出力しており、行アドレス・コントロール・クロックの出力と論理をとり、必要な時にメモリに制御信号を与えます。

## フィールド・メモリの実際の動作

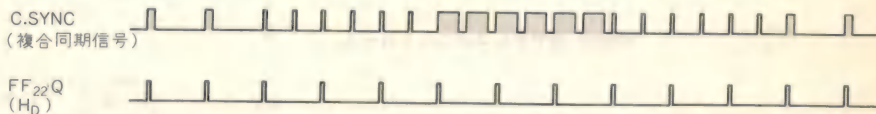
### ●ライト・モード(画像記憶)

NTSCコンポジット信号の場合、1フレームはイン

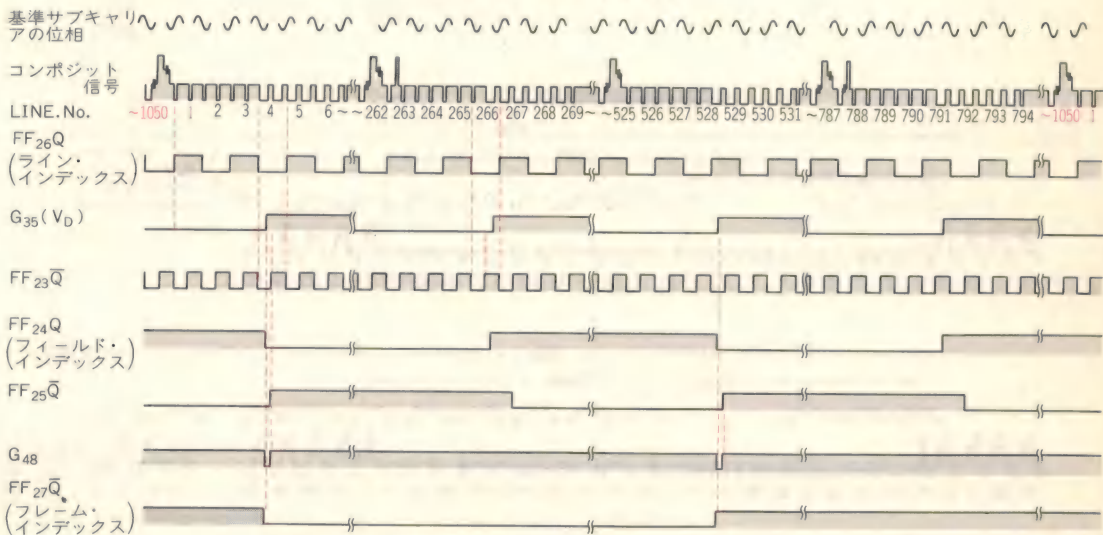
〈図17〉  
水平同期信号  
H<sub>0</sub>発生回路



〈図19〉  
水平同期信号H<sub>0</sub>



〈図20〉 フレーム・インデックス



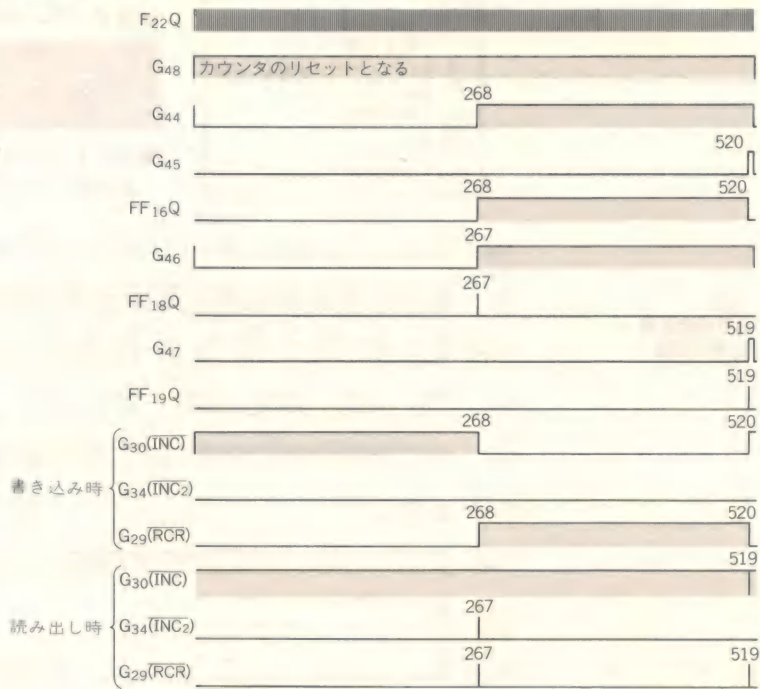


タレース走査により2フィールドで構成されています。したがって、1フレームを1単位として1番目から273番目までと、その直前のフレームの525番目のラインをメモリに記憶させます。そして図18に示したように、第1フィールドと第2フィールドの垂直同期期間

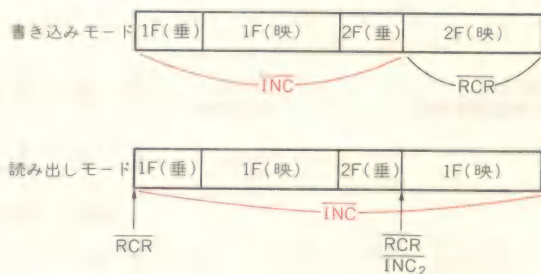
を、メモリ・セル上に構成させます。

ここで第2フィールドの垂直同期期間を記憶するのは、読み出しの時に第1フィールドの信号だけで連続して読み出すと、各フィールド間で、図25のように0.5Hのスキュが発生するからです。

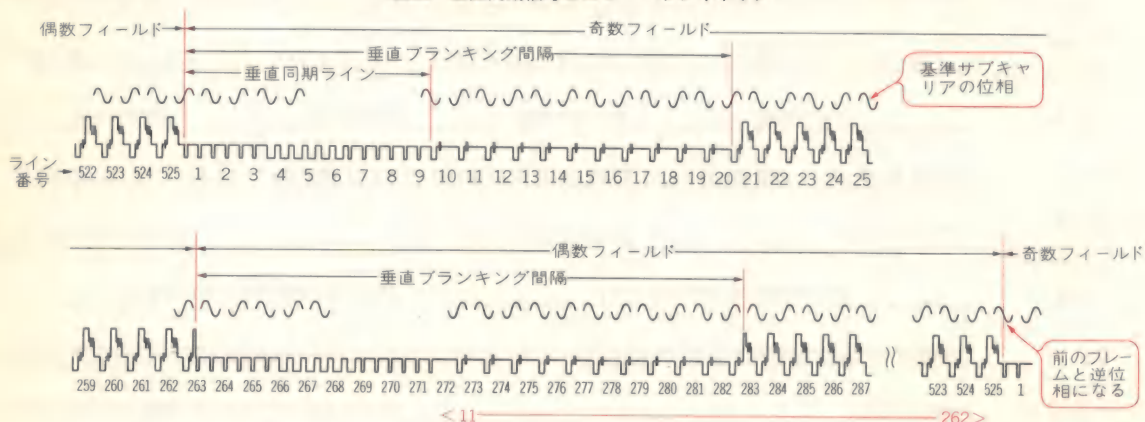
〈図21〉 行アドレス・コントロール・クロック



〈図22〉 行アドレスのコントロール



〈図23〉 垂直同期信号とカラー・サブキャリア





また525番目のラインの記憶は、1ライン中で実際にメモリに記憶している部分は、図26に示す部分だけです。そのため、1番目のラインより記憶した場合の、1番目のラインの同期信号の立ち下がり部分の記憶情報が欠けることを防いでいます。

この処理は、通常の動作上はどのラインの水平同期信号も同位相であるため、特に直前フレームの525番目のラインを記憶する必要はありません。ほかのラインのものでも代用は可能ですが、ここではフィールド・メモリを前提として、あえて記憶しています。

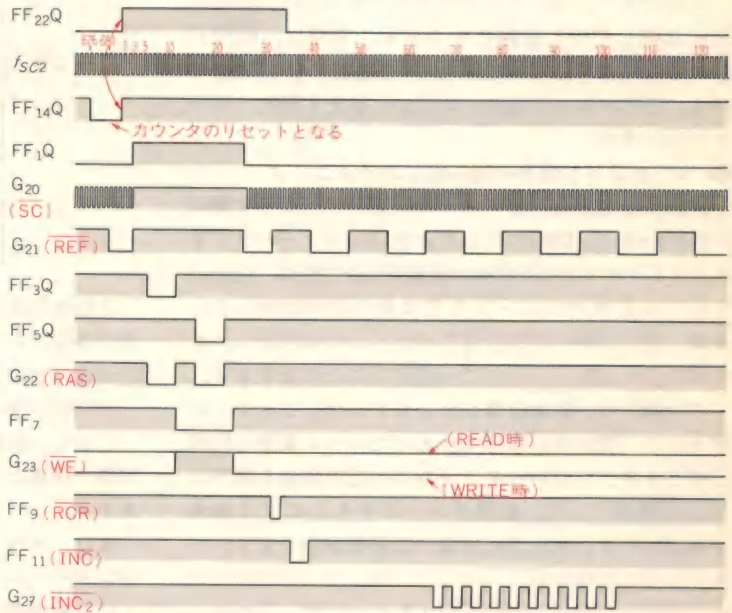
実際のメモリ・コントロール、特に行アドレスの指定については、記憶させるラインでは直前の1ライン

期間に $\overline{\text{INC}}$ パルスを与え、ラインごとに順次アドレスを進めます。そして記憶させないラインに関しては、直前の1ライン期間に $\overline{\text{RCR}}$ パルスを与えることにより、行アドレス・カウンタをリセットしています。

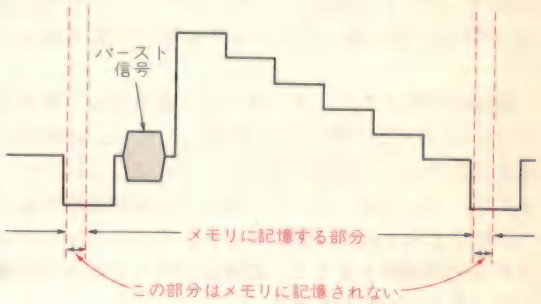
● リード・モード(静止画表示)

メモリに記憶した情報を読み出す時は、図27に示すように、第2フィールドの映像信号期間は、メモリに記憶した第1フィールドの信号で補います。そして、垂直同期期間に関しては、第1フィールドと第2フィールドのものを交互に読み出すようにしています。こうすることにより、読み出した信号をモニタ出力とし

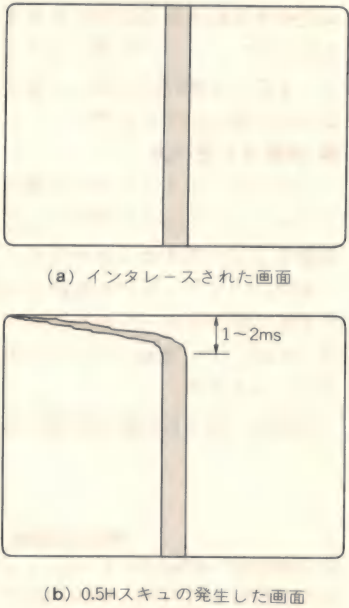
〈図24〉メモリ・コントロール・クロック



〈図26〉記憶情報(メモリに記憶する部分)



〈図25〉0.5Hスキュ



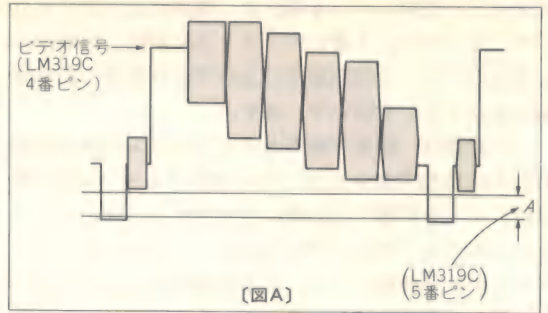
〈図27〉メモリ・セルの読み出し方式

第1フィールド		第2フィールド	
1F(垂)	1F(映)	2F(垂)	1F(映)



〈表2〉 調整方法

調整点	調整方法
図14 VR <sub>1</sub>	LM319Cの5番ピンのDC電位が図Aに示すAの範囲(できる限り中心が望ましい)にくるように調整する。
図14 VR <sub>2</sub>	$\mu$ PC1536Cの16番ピンのSWをONした時に(通常はOFF), $40f_H$ 信号の周波数が, $629\text{kHz} \pm 5\text{kHz}$ になるように調整する。
図15 VR <sub>3</sub>	入力ビデオ信号を白100%信号とした時に、ビデオ出力端子のレベルが2V <sub>P-P</sub> になるように調整する。
図14 VR <sub>4</sub>	垂直同期信号V <sub>D</sub> パルスが、写真8に示す波形となるように調整する。



〔図A〕

(LM319C  
5番ピン)

て、インタレース走査を可能としています。

そのために、行アドレスのコントロールは、記憶の時に比較して少し複雑になります。実際は、RCRパルスと $\text{INC}_2 (= \text{INC} \times 10)$ パルスを図22に示す位置に与え、 $\text{INC}$ パルスは各ラインごとに与えて順次アドレスを進めます。

### ● コンポジット信号の読み出しのテクニック

NTSC信号では、1フレームは2フィールドで構成されています。図23に示したように、カラー・サブキャリア(バーストと同位相の連続波)の位相は、最初のフレームと次のフレームとでは180°位相反転しています。そのため、同期信号とカラー・サブキャリアの位相関係は4フィールドで一巡することになります。

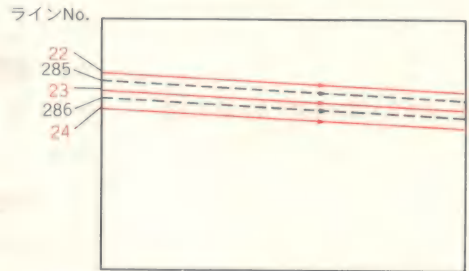
この関係は、映像信号をデジタル化してメモリに書き込み、読み出しを行ううえでたいへん重要な問題となります。したがって、メモリの記憶情報を図19に示した構成にして、基準クロックを $3f_{sc}$ にする場合を考慮すると、次の点に注意する必要があります。

▶メモリに記憶されているラインは、1～272番目とその直前フレームの525番目のラインです。したがって、読み出しの時に1フレームを構成するためには、記憶時にメモリに記憶していないライン(273～524番目)に関しては、第1フィールドの信号で置き換えます。

図28に示すように、インタレース走査では、例えば285番目のラインの置き換え信号としては、22番目か23番目のラインの画素情報をもっとも適しています。しかし、ここでカラー・サブキャリアは連続波でなければなりませんので、図23に示したタイムチャートよりその位相関係を見ると、23番目のラインのほうが適していることがわかります。

▶前記の操作により、1フレーム分の信号は構成できます。しかし、NTSC信号では、フレーム間でカラー・サブキャリアの位相は反転していますので、読み出しをそのまま続けると、フレーム間でカラー・サブキャリアの連続性がなくなることになります。この場合、実際のモニタ出力では画面上部で色相が回り、たいへん見苦しくなります。

〈図28〉 インタレース走査



この点の防止策としては、読み出し信号のカラー信号のみをフレームごとに位相反転する、あるいは、フレームごとに信号そのものをカラー・サブキャリアの1/2周期分だけ位相をずらすことなどが考えられます。

しかし、前者はY/C分離などの処理が必要であり、回路が複雑になりますので、ここでは後者を用いています。

また、メモリに記憶した信号を読み出す時のサンプリング・クロック $\overline{\text{SC}}$ については、これを常時入力されている映像信号から $\overline{\text{SC}}$ を作る場合は、フレームごとに位相を反転させる必要があります。というのは、書き込んだフィールドを異なるフレーム側の信号でクロックを作って読み出す時に、書き込み時と同位相となる条件で読み出すためです。

### ● 調整法と使い方

このフィールド・メモリの操作は簡単です。図16のリセット・スイッチを押すことにより、その時点での画像をフリーズすることができます。

また、パソコンなどと接続して使う場合には、図15のメモリ出力をポートを介してパソコン側メモリに転送すれば、データをパソコンで処理することができるようになります。

最後に、表2に回路の調整方法を示します。

### ● 参考文献

- (1) 日本電気,  $\mu$ PD41221Cデータ・シート。
- (2) 日本電気,  $\mu$ PC1536C,  $\mu$ PC2310Cデータ・シート。



## 倍速スキャン・コンバータの設計・製作



湯川達雄

テレビの水平走査周波数を倍速にしてノンインタレースにする  
と、画面が塗りつぶしたようにくっきりとなります。ここでは高  
速ライン・メモリを使用した具体例を紹介します。

### インタレース→ノンインタレース変換の効果

#### ● インタレース走査とは

テレビやVTRなどの映像信号は、画面のちらつきを少なくするためにインタレース(飛び越し)走査を採用しています。すなわち、図1のように走査線を1本おきに走査して垂直方向の画素密度を下げる代わりに、フィールド周波数(1秒間当たりの画面の数)を2倍にする方式をとっています。

インタレース走査では、第1フィールドと第2フィールドとで1枚の完全な絵(フレーム)ができあがります。したがって、NTSC方式のテレビの場合、一つのフィールドの走査線数は262.5本で、画面数は30枚/秒となっています。

このように、インタレース走査では1画面当たりの垂直方向の画素密度(走査線数)を下げていますので、走査が粗くなり、走査線と走査線の間のすき間が見え

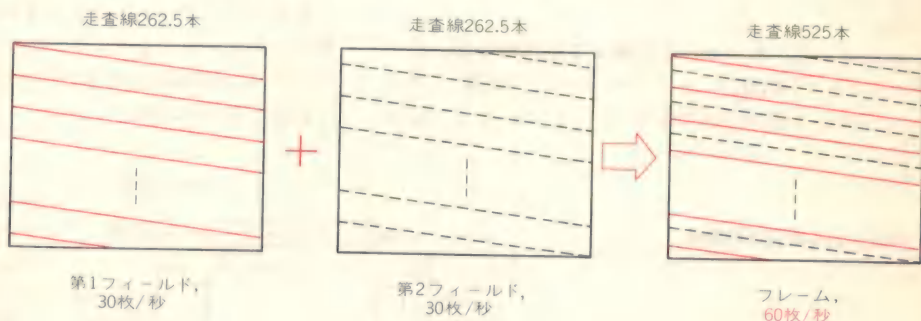
ることになります。これは画面が大きいくほど顕著になり、見づらい画面になります。

#### ● スキャン・コンバータとは

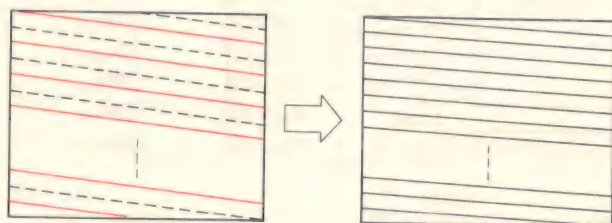
このインタレース走査による垂直方向の画素密度の劣化を解消するには、1走査線単位での信号を2倍に圧縮します。そして、フィールド周波数(垂直周波数)はそのままで水平周波数を2倍にして、1フィールド当たりの走査線数を2倍にする方法があります。これによりインタレース走査がノンインタレース走査に変換され、垂直方向の画素密度を上げることができます。このようすを図2に示します。

このインタレース→ノンインタレース走査変換を実現するためには、1走査線ごとにメモリに信号を書き込み、書き込みの2倍の速度で読み出せばよいことになります。これがスキャン・コンバータと呼ばれるものの原理です。このようすを図3に示します。

〈図1〉  
インタレース走査のしくみ



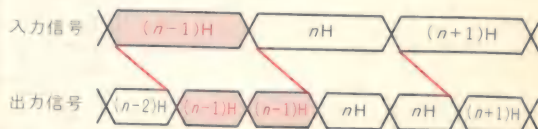
〈図2〉インタレース走査とノンインタレース走査の違い



インタレース走査

ノンインタレース走査

〈図3〉スキャン・コンバータの原理





## 画像用高速ライン・メモリ

前述のようなスキャン・コンバータを実現するには、1走査線ごとのライン・メモリが必要となりますが、これに適したメモリが $\mu$ PD41101Cです。

### ● $\mu$ PD41101Cの働き

この $\mu$ PD41101Cは、910ワード×8ビット構成の高速ライン・メモリです。NTSC方式の映像信号をカラー・サブキャリアの4倍の周波数(14.3MHz)のサンプリング・レートで8ビットに量子化した場合に、

$f_{sc}$ : カラー・サブキャリア周波数

$f_H$ : 水平周波数

とすると、

$$4f_{sc} = 4 \times \frac{455}{2} f_H = 910f_H$$

となり、ちょうど1ライン分(1水平走査線)が記憶できる構成になっています。

したがって、映像信号をディジタル処理するのに最適であり、1ライン遅延、時間軸変換などが容易に実現できます。

図4に $\mu$ PD41101Cのブロック図と端子接続を示します。このICは速度により表1のようにファミリがあります。今回は-2を使用します。

$\mu$ PD41101Cはアドレス・ポインタを内蔵していますので、外部にアドレス発生回路は不要です。また、ライト/リードが独立で、かつ非同期に行うことができますので、汎用のスタティックRAMを使用する場合と異なり、ダブル・バッファ構成にする必要はありません。

したがって、本メモリを制御する外部信号は、ライト・クロック(WCK)、リード・クロック(RCK)、ライト・リセット信号(RSTR)、リード・リセット信号

〈表1〉 $\mu$ PD41101Cファミリ構成

ファミリ	リード・サイクル時間	アクセス時間	ライト・サイクル時間
$\mu$ PD41101C-3	34 ns	27 ns	34 ns
$\mu$ PD41101C-2	34 ns	27 ns	69 ns
$\mu$ PD41101C-1	69 ns	49 ns	69 ns

(RSTR)の4種類だけとなります。

### ● タイミングの説明

ライト・サイクルは、図5に示すように $\overline{WE}$ が“L”レベルのとき、WCKに同期して1サイクルに1アドレス(8ビット=1バイト)ずつ行われます。

ライト・アドレスは、ライト・サイクルが終了するたびに自動的に1アドレスをインクリメントします。したがって、ライト・データは、WCKの立ち上がりエッジに対して、セットアップとホールド・タイムを満足するように加える必要があります。

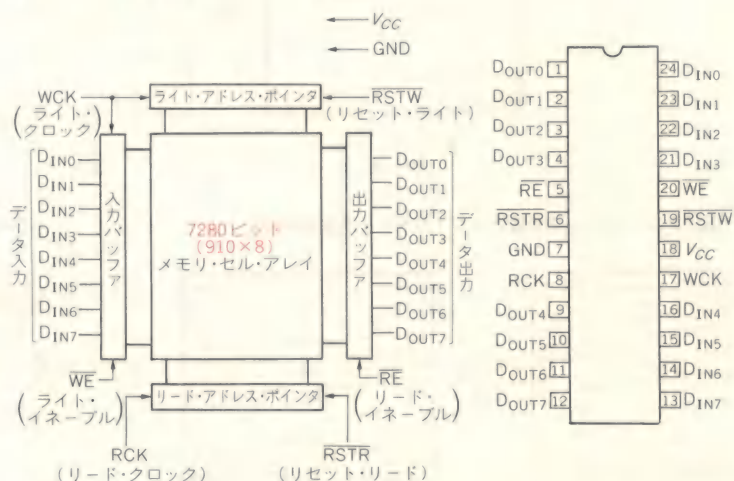
RSTW信号は図6に示すように、ライト・アドレス・ポインタを0アドレスにリセットするための信号入力です。WCKの立ち上がりエッジに対して、セットアップとホールド・タイムを満足するように入力します。

ライト・アドレス・ポインタはWCKに同期して一つずつインクリメントされ、最後のアドレスまでくると、次のアドレスは自動的に0にもどります。

$\overline{WE}$ が“H”レベルのときはデータは入力されず、WCKに無関係にライト・アドレスも停止します。

リード・サイクルは図7に示すように、 $\overline{RE}$ が“L”レベルのとき、RCKに同期して1サイクルに1アドレス(8ビット=1バイト)ずつ行われます。リード・アドレスは、リード・サイクルが終了するたびに、自動的に1アドレスをインクリメントします。リード・データは、RCKの立ち上がりエッジからアクセス・タイム後に出力されます。

〈図4〉 $\mu$ PD41101Cの構成



(a) ブロック図

(b) ピン接続



**RSTR**信号は、リード・アドレス・ポインタを0アドレスにリセットするための信号入力です。図8に示すようにRCKの立ち上がりエッジに対して、セットアップとホールド・タイムを満足するように入力します。リード・アドレス・ポインタは、RCKに同期して一つずつインクリメントされ、最後のアドレスまでくると、次のアドレスは自動的に0にもどります。

**RE**が“H”レベルのときはリードが禁止され、出力がハイ・インピーダンスになると同時に、RCKに無

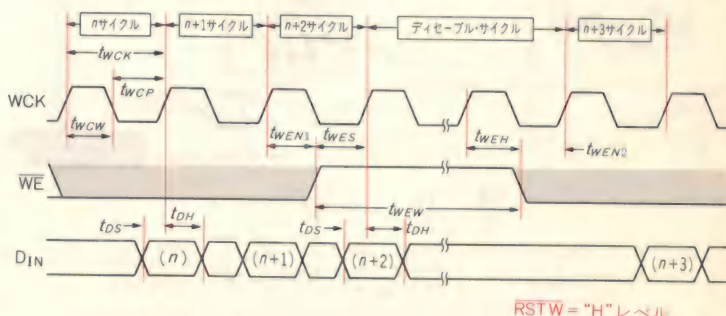
関係でリード・アドレスも停止します。図8がリード・リセット・サイクルです。

## スキャン・コンバータの構成方法

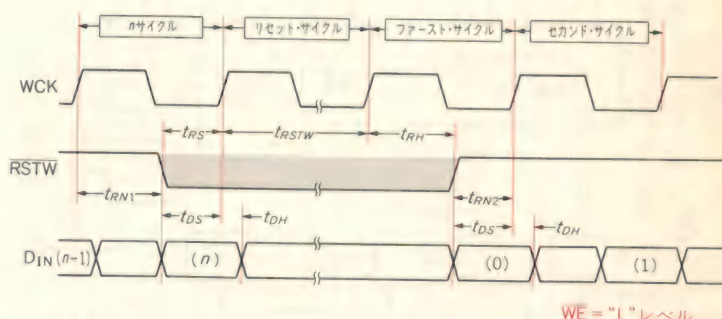
### ● ビデオ信号入力部の構成

図9にスキャン・コンバータのブロック図を示します。信号入力としては**NTSCコンポジット信号**と**RGB入力**とがあります。NTSCコンポジット信号入

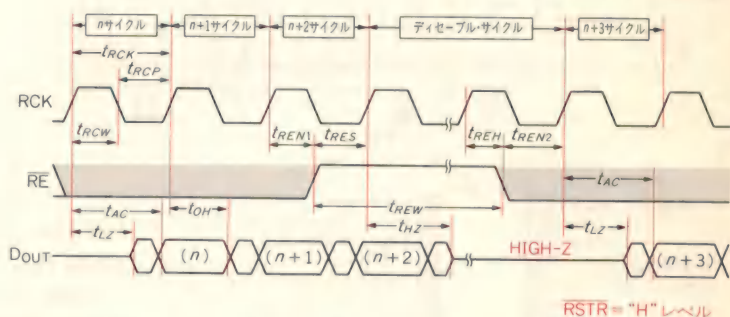
〈図5〉  
μPD41101Cのライト・サイクル



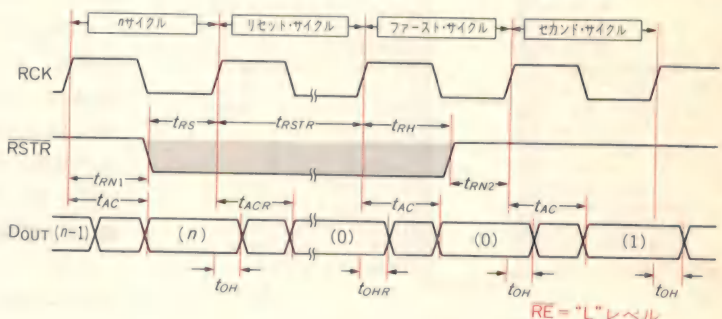
〈図6〉  
μPD41101Cのライト・リセット・サイクル



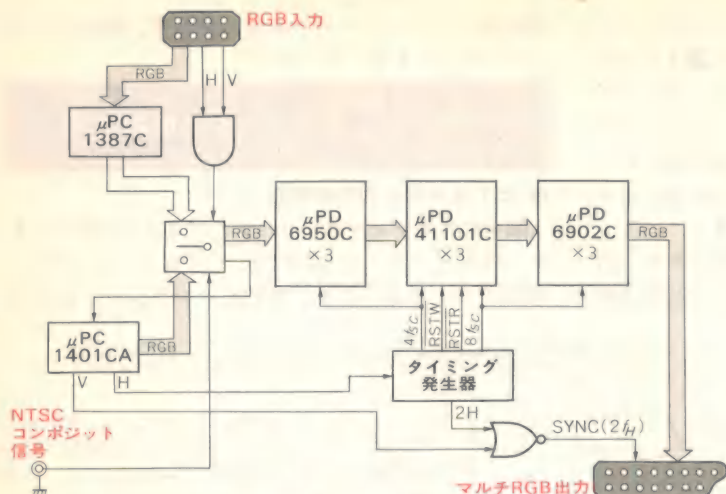
〈図7〉  
μPD41101Cのリード・サイクル



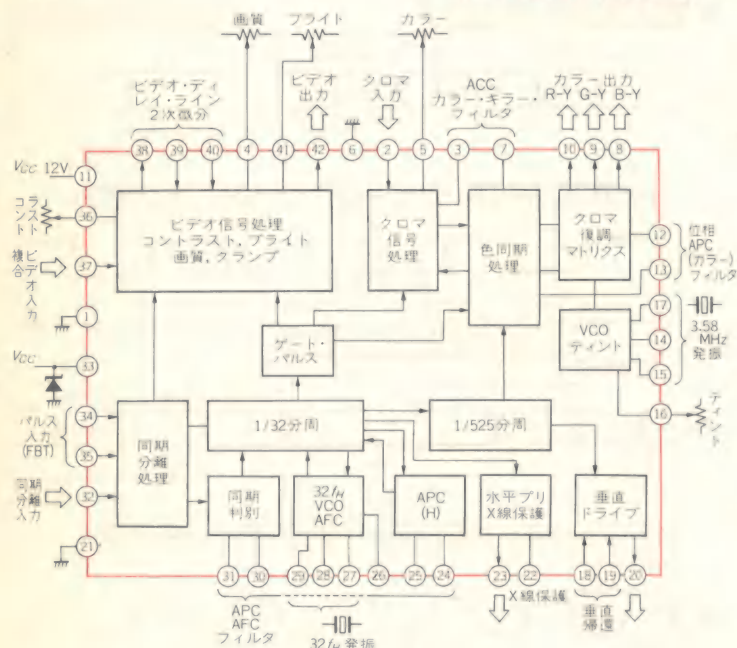
〈図8〉  
μPD41101Cのリード・リセット・サイクル







〈図9〉 スキャン・コンバータのブロック図



〈図10〉 μPC1401CAのブロック図と端子接続

(a) ブロック図

接地(21ピン以外)	1	42	ビデオ出力
クロマ入力	2	41	ブライト・コントロール
コンデンサ(ACC)	3	40	画質(2次微分)入力
画質コントロール	4	39	ビデオ入力
カラー・コントロール	5	38	ビデオ、クロマ出力(コントラスト)
接地	6	37	ビデオ、クロマ入力
カラー・キラー・フィルタ	7	36	コントラスト・コントロール
B-Y出力	8	35	プライバック・パルス入力
G-Y出力	9	34	水平フランクング入力
R-Y出力	10	33	電源(水平)
電源(33ピン以外)	11	32	ビデオ入力(同期分離)
位相フィルタ	12	31	水平キラー・フィルタ
APCフィルタ(クロマ)	13	30	水平キラー・スイッチ出力
カラー・VCOフィルタ	14	29	水平VCOフィルタ
カラー・VCOフィルタ	15	28	水平VCOフィルタ
ティント・コントロール	16	27	水平VCOフィルタ
カラー・VCO発振素子	17	26	ローパス・フィルタ(f)
垂直帰還入力(DC)	18	25	可変位相フィルタ(水平)
垂直帰還入力(AC)	19	24	ローパス・フィルタ(φ)
垂直ドライブ出力	20	23	水平プリドライブ出力
接地(水平、垂直、クロマ、OSC)	21	22	X線プロテクト入力

(b) ピン接続

力端子は、テレビやVTRのビデオ出力端子に、RGB入力端子は、パソコンの出力端子に接続します。

NTSCコンポジット信号は、μPC1401CAにより復調され、RGB信号となります。

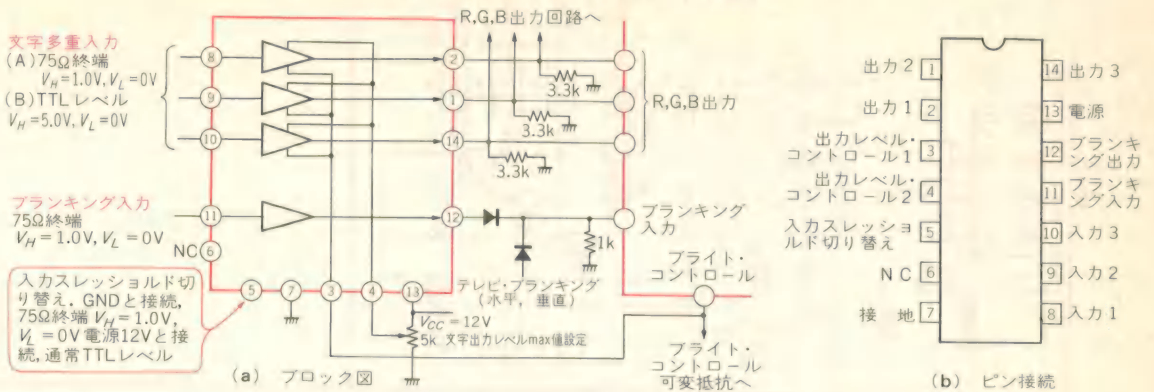
このμPC1401CAは、NTSC方式カラー・テレビの色信号、ビデオ信号、同期信号処理用ICです。色同期(APC)、水平同期(H-HOLD)、垂直同期(V-HOLD)の完全無調整化が実現でき、周辺部品点数および調整の簡素化が実現可能です。図10に構成と端子接続を示します。

#### ● RGB信号のA-D変換

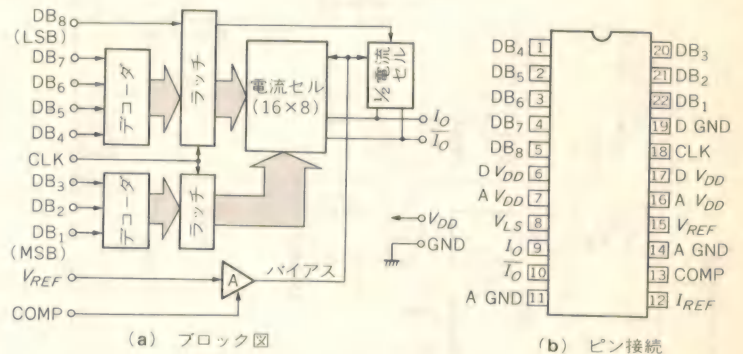
パソコンからのRGB信号は、μPC1387Cによりレベル合わせをします。そして、二つのRGB信号を切り



〈図11〉  $\mu$ PC1387Cのブロック図と端子接続

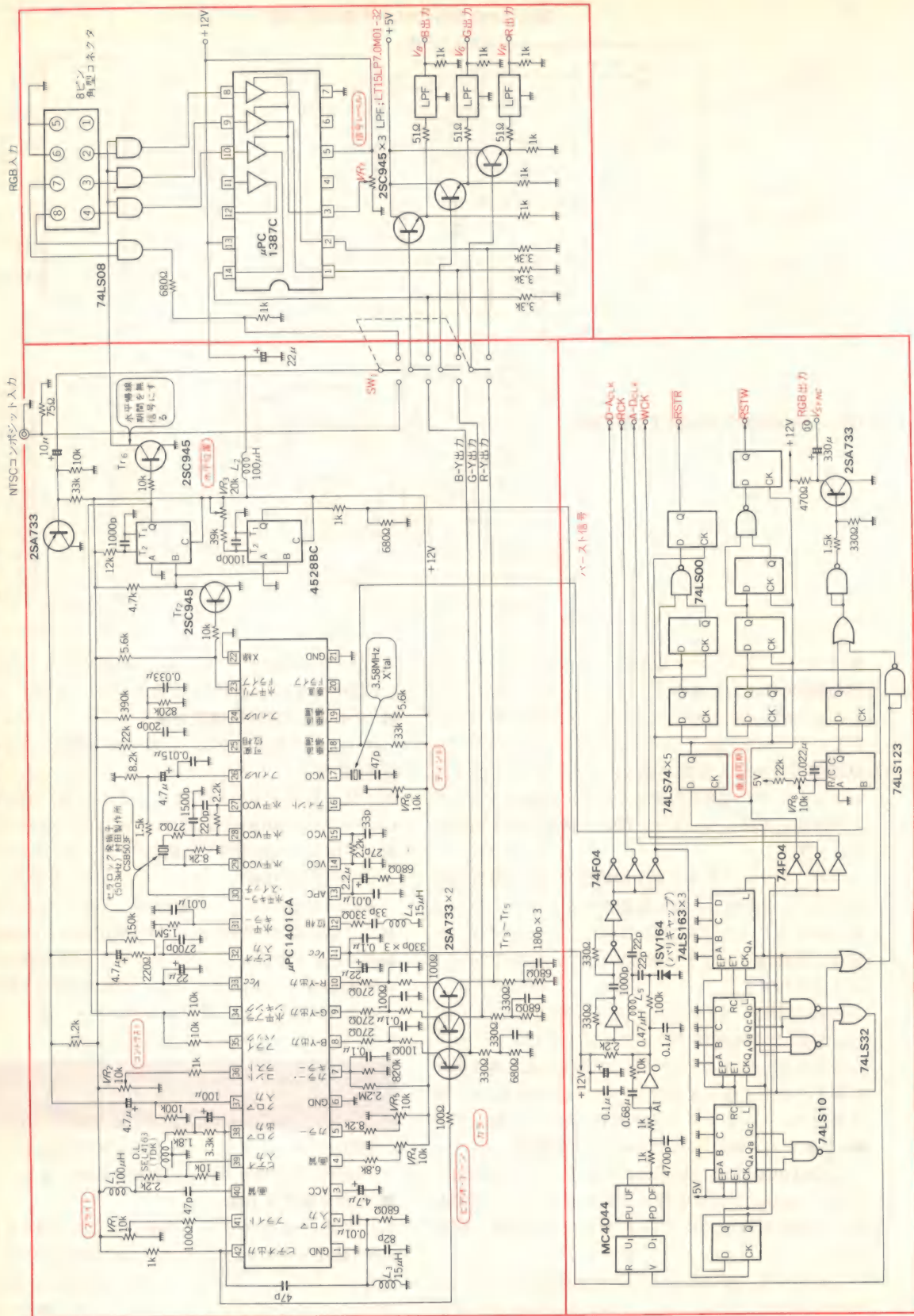


〈図12〉  $\mu$ PD6902Cのブロック図と端子接続



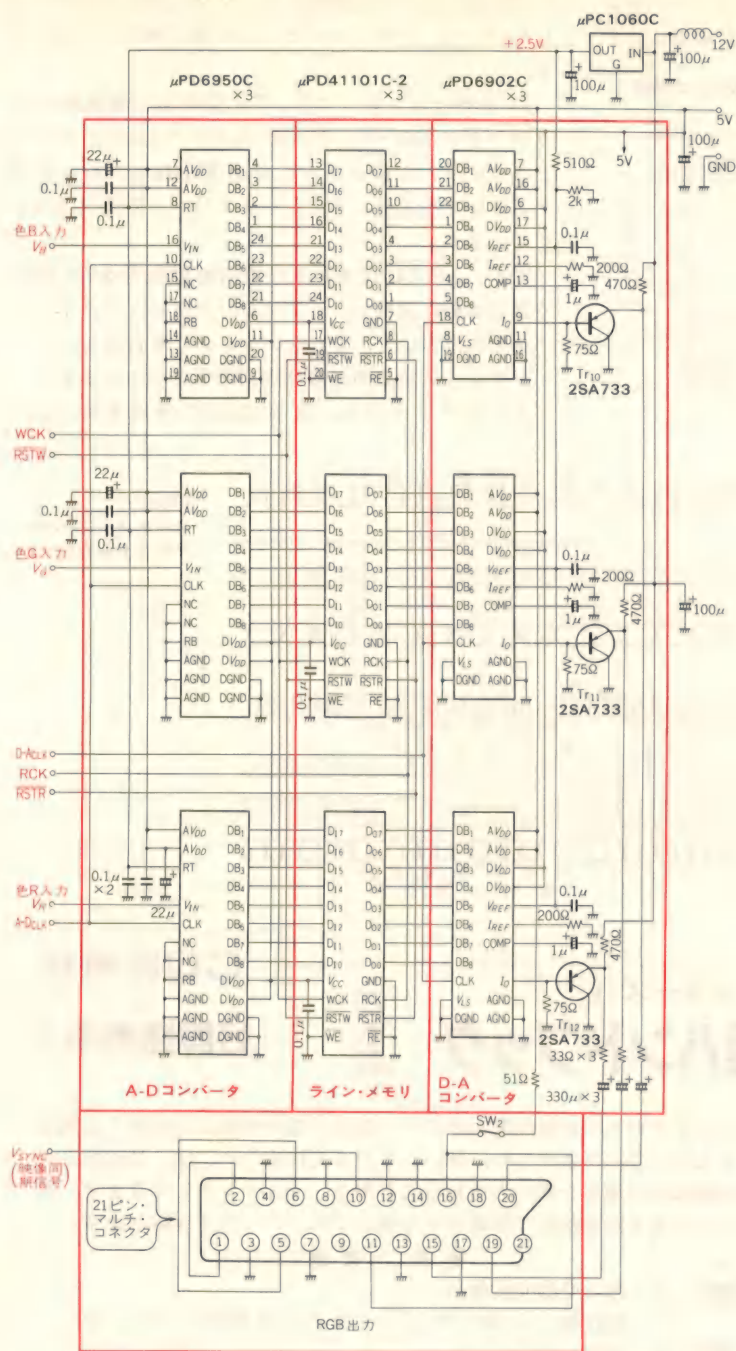


〈図13(a)〉 スキャン・コンバータ回路





〈図13(b)〉 スキャン・コンバータ回路(つづき)



した水平駆動信号, および垂直駆動信号を出力します。

このICは色差出力形式(Y, R-Y, B-Y, G-Y)ですので, 外部でトランジスタ3本( $Tr_3 \sim Tr_5$ )を使用し, マトリクスを構成してRGB信号としています。

なお, この部分のワンショット・マルチバイブレータ4528BCは, 水平駆動信号を適当なパルス幅にするために使用しています。一つはバースト・ゲート・パ

ルス, およびブランキング・パルスとして,  $\mu PC1401CA$ の34番, 35番ピンに入力します。もう一つは, クロック発生器の位相比較用として, 位相比較用IC MC4044へ入力します。

### ● RGB信号入力部

8ピン角型コネクタより入力されたRGB信号は, 74LS08を経て $\mu PC1387C$ に入力されます。74LS08および $Tr_6$ は, 水平帰線期間を無信号にするためのものです。

$\mu PC1387C$ では, 入力されたRGB信号を適正なレベルに調整して出力します。一方, 水平駆動信号と垂直駆動信号は74LS08で合成され複合同期信号となり, スイッチ( $SW_1$ )により選択されて $\mu PC1401CA$ に入力されます。

### ● A-Dコンバータ部

スイッチ( $SW_1$ )により選択されたRGB信号は, ローパス・フィルタ(LPF)を経て $\mu PD6950C$ へ入力されます。このローパス・フィルタは7MHzのもので, サンプリング周波数(14.3MHz)の1/2以上の周波数をカットするために使用します。図14にこのフィルタの特性を示します。

入力されたRGB信号はそれぞれ14.3MHzのクロックでA-D変換され, 8ビットのデジタル信号として $\mu PD41101C$ に入力されます。

### ● ライン・メモリとD-Aコンバータ

入力された8ビットのデジタル信号を14.3MHzで書き込み, 2倍の28.6MHzで読み出しながらD-Aコンバータ $\mu PD6902C$ へ入力します。

メモリのコントロール信号(WCK, RCK, RSTW, RSTR)はタイミング発生器から供給されます。これらのタイムチャートを図15に示します。

ライン・メモリより入力されたデジタル信号を28.6MHzのクロックでD-A変換し, 2倍に圧縮されたRGB信号に再生します。

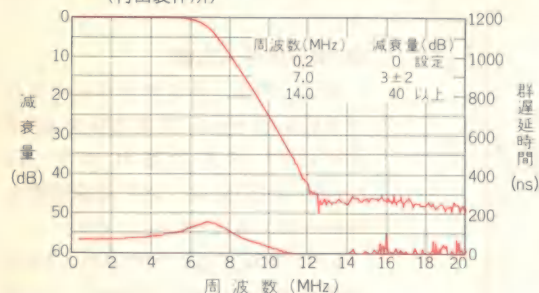
### ● タイミング発生部

74F04を使用したLC発振回路により, 28.6MHzのクロックを発生します。このクロックは, そのままライン・メモリおよびD-Aコンバータの駆動用クロック



として使用します。また、このクロックを74LS74により1/2に分周(14.3MHz)して、A-Dコンバータおよびライン・メモリの駆動用クロックとして使用します。

〈図14〉 ローパス・フィルタ、LT15LP7、OM01-32の特性  
(村田製作所)

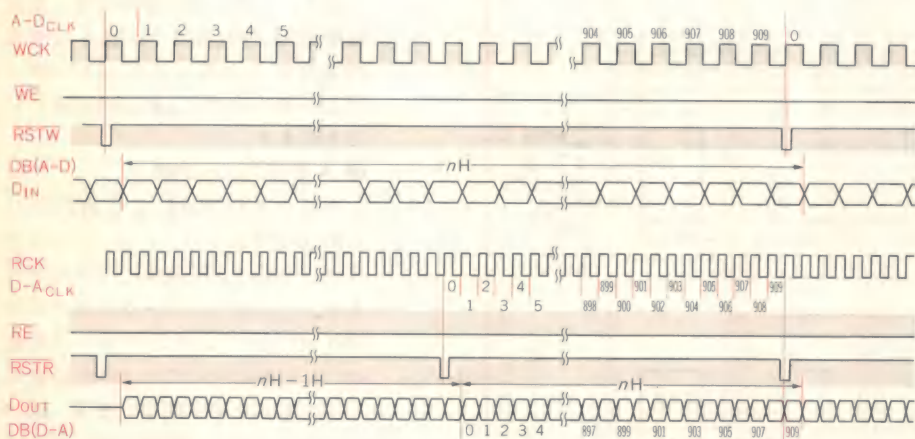


なお、この回路の部分は第5章で示したスーパインポーズ回路での水平同期型クロック発生回路と同じです。したがって、タイミングとしては第5章図7と同一です。

14.3MHzのクロックは、74LS163を3個使用したカウンタで455分周され、**31.5kHz ( $2f_H$ )**のクロックとなります。このクロックを28.6MHzのクロックでタイミングをとり、**ライン・メモリへのRSTR信号**としています。

また、 $\mu$ PC1401CAからの垂直駆動信号を74LS123で適当な幅に調整し、この $2f_H$ のクロックでタイミングをとったものと、 $2f_H$ のクロックとを合成し、ノンインタレース走査用の複合同期信号としています。

$2f_H$ のクロックは、さらに1/2に分周され、14.3



〈図15〉 スキャン・コンバータのタイムチャート

— ハードウェア・デザイン・シリーズ ① —

# 電子回路部品活用ハンドブック

2色刷

CQ出版社

好評発売中!!

B5判 288頁 定価 1,850円

## 電子回路部品活用ハンドブック

受動部品、機構部品を100%活用するために



エレクトロニクス化が進む中でIC周辺の電子回路部品に対する関心も大きくなってきました。良いシステムを作るためには、受動部品/機構部品の適切な使い方を知ることがキー・ポイントとなります。本書では電子回路部品の基礎から応用までのノウハウを詳解します。

### 目次

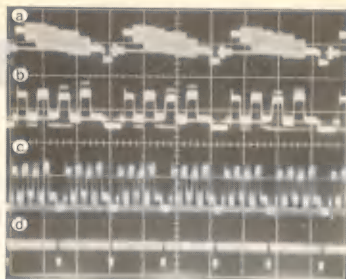
- 第1章：基本受動回路部品  
抵抗器/コンデンサ/インダクタ/水晶振動子/バラン他
- 第2章：機構的電子回路部品  
コネクタ/スイッチ/感温リード・スイッチ/同軸リレー他
- 第3章：電源回路部品  
トランス/電池/サージ・アブソーバ/パワー・サーミスタ他
- 第4章：ノイズ対策用部品  
EMIフィルタ/パスパ/導電性塗料他
- APPENDIX  
部品の信頼性/部品のアートワーク図



### 〈写真1〉

#### 各部の動作波形

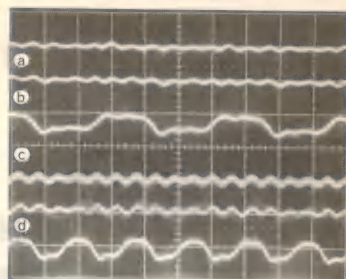
(a): NTSC コンポジット信号, (b): 復調されたB信号, (c): スキャン変換後のB信号, (d): ノンインタレース走査用複合同期信号)



### 〈写真2〉

#### 各部の動作波形

(a): ライン・メモリへのデータ入力, (b): ライト・クロック, (c): ライン・メモリからのデータ出力, (d): リード・クロック)



MHzのクロックでタイミングをとり、ライン・メモリへのRSTW信号としています。また、クロック発生器の位相比較用としてMC4044へも入力します。

$\mu$ PC1401CAからの水平信号と、クロック発生器からのクロックを分周して得られた水平信号とをMC4044で位相比較し、ローパス・フィルタを通してクロック発生器(VCO; 電圧制御発振器)のバラクタ・ダイオード(1SV164)に印加し、発振周波数を制御します。

### ● RGB出力部

ノンインタレース化されたRGB信号と複合同期信号を、21ピン・マルチ・コネクタによりテレビ・モニタに出力します。

テレビ・モニタは、水平走査周波数が31.5kHzで動作可能なものを使用します。この種のモニタとしてはPC-TV451[日本電気ホームエレクトロニクス(株)]やKX21HD1, KX14D1[ソニー(株)]などがあります。切り替えスイッチ(SW<sub>2</sub>)は、モニタの表示モードの切り替え用(外部/内部)です。スイッチがONの時に外部表示モードとなり、ノンインタレース化された画面がモニタに表示されます。

### ● 調整方法

#### ▶ $\mu$ PC1401CA周辺

ブライトネス( $VR_1$ )、コントラスト( $VR_2$ )、カラー( $VR_3$ )は、A-Dコンバータ  $\mu$ PD6950Cの入力で見ても、信号の振幅が最大で0~2.5Vの範囲になるように調整します。

ビデオ・トーン( $VR_4$ )は好みの画質に合わせてください。また、ティント( $VR_5$ )は適正な色相が得られるように調整してください。カラー( $VR_6$ )およびティント( $VR_7$ )は調整用つまみをセットにつけてもよいと思

います。

$VR_3$ は水平位置調整用です。適当な位置が得られるように調整します。

#### ▶ $\mu$ PC1387C周辺

$VR_7$ は信号レベルの調整用です。A-Dコンバータ  $\mu$ PD6950Cの入力で見ても、信号レベルが2~2.5V程度になるように調整します。

#### ▶ タイミング発生器周辺

$VR_8$ は垂直同期出力信号の幅を調整するものです。水平クロックのエッジによって垂直同期信号がばたつかないように、74LS123の出力で見ても、垂直同期信号の立ち下がりエッジが、水平クロック(74LS74のクロック入力)の立ち上がりエッジと立ち上がりエッジの中間あたりになるように調整します。

写真1、写真2に実際の各部の動作波形を示します。

### ● 製作上の注意点

このシステムは、ディジタル回路とアナログ回路との混在になっていますので、基板製作には特に注意が必要です。

第9章で紹介したフィールド・メモリと同等な慎重な注意が必要です。特にA-D/D-Aコンバータ周辺では、アナログ・グラウンドとディジタル・グラウンドとの分離が大切です。

### ● 参考文献 ●

- (1) 日本電気,  $\mu$ PD41101Cデータ・シート。
- (2) 日本電気,  $\mu$ PC1401CA,  $\mu$ PC1387Cデータ・シート。
- (3) 日本電気,  $\mu$ PD6902Cデータ・シート(開発速報)。



## パソコンによる画像処理技術



桑村伸一  
坂巻 顯

カメラで捉えた画像を、パターン認識などに利用するのにパソコンを利用する例が増えています。ここでは市販のパソコン用画像入力ボードを使い、画像処理技術の実情について検討しています。巻頭のカラー口絵も参照してください。

最近のメモリ素子の大容量化に伴い、画像処理システムが身近になってきました。特にパソコンの処理速度の向上に伴い、パソコンで画像処理を実用的に行いたいという要望が出てきました。

また、パソコンの拡張スロットに差し込んで数画面分のメモリ容量をもった、画像処理用のボードが各社から販売されています。ここではこのようなボードを利用して、どのようなことができるかをわかりやすく解説したいと思います。

### 画像処理とは

#### ● 用途から考えると

画像処理の用途は非常に広範囲にわたり一概に決まりません。パソコンで行う場合は価格が安いために、実際に画像処理をして役立てようという場合の**基礎的な実験**が簡単にできます。しかし、実際にシステムに組み込んで利用している例も多くあります。

画像処理を行う目的として一番多いのが、**目視検査を自動化する**用途です。この用途では検査しようとする物の個数、長さ、面積、重心といったデータを画像から求めます。もっと単純化すれば、画面中のあるエリアに、検査する物があるかないかということをチェックすることもあります。

工業用途では、処理は簡単でもスピードを要求する場合があります。その場合はシステムのスピードを最初に検討しておくことが重要です。

そのほか医学用途も以前から研究されています。医学用途では細胞の顕微鏡画像をテレビ・カメラで取り込み、画像処理を行って検査します。また、**CT画像や超音波断層像から断層像を多数取り込み、3次元に再構成し立体像にするシステム**もあります。

今後、画像処理を応用したシステムが多くなるものと思われます。

#### ● 2値化がベース

今回実験する画像処理用メモリ・ボードは、表1のような仕様になっています。

この画像処理ボードの説明は、後述のパソコン画像

処理システムの構成法で詳しく解説します。**写真1**にその外観を示します。

写真2が実験に使ったシステムの全景です。CCDカメラ、PC9801VX2、マルチスキャン・カラー・モニタ・テレビを使いました。モニタ・テレビは白黒モニタ・テレビでもよいのですが、メモリ・ボードから疑似カラーが出力できますのでカラーにしました。

写真3が処理される入力の前画像の一例で、写真4が写真3の前画像を2値化した画像です。

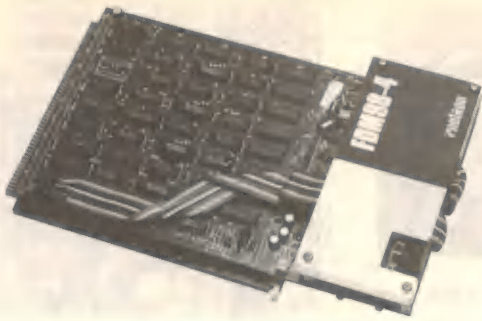
メモリ・ボードのデータは6ビット(64階調)ですので、写真4は階調がちょうど中間値で2値化されています。中間値より明るいと白、暗いと黒の2色の画像に変換します。平面的な被写体の場合は、この例のように照明が均等に当たっている必要があります。

2値化して白または黒の**画素の数を数えると面積が求められます**。この面積はたんに画素数にすぎませんので、長さがわかっている既知のものでキャリブレーション

〈表1〉画像処理ボードFDM98-4の仕様〔㈱フォトロン〕

入力信号：日本標準テレビ信号	
水平周波数	15.75kHz
垂直周波数	60Hz
映像レベル（複合映像信号）	1V <sub>p-p</sub> 75Ω
出力信号	
映像レベル（複合映像信号）	1V <sub>p-p</sub> 75Ω
擬似カラー映像信号	
RED(TTL)	正論理
GREEN(TTL)	正論理
BLUE(TTL)	正論理
H. SYNC(TTL)	負論理
画像メモリ容量	
256×256×8×4	
合計256Kバイト	
A-Dコンバータ	
6ビット（64階調）	
基準電圧コントロール	
（+）側電圧 256段階に設定可能	
（-）側電圧 256段階に設定可能	
D-Aコンバータ	
8ビット（256階調）	
画素数	
横	256画素
縦	256画素





〈写真1〉 画像処理ボードFDM98-4の外観



〈写真2〉 システムの全景



〈写真3〉 入力原画像



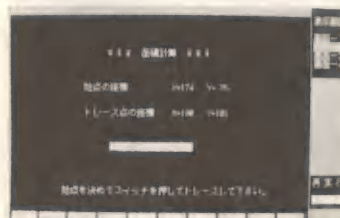
〈写真5 (a)〉 トレース



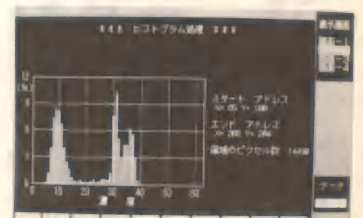
〈写真6 (a)〉 カーソル内側のヒストグラムを求める



〈写真4〉 2値化像



〈写真5 (b)〉 面積の結果表示



〈写真6 (b)〉 ヒストグラムの結果表示

ションしておく必要があります。キャリブレーションした係数と画素数をかけて、実際の面積を求めます。

2値化する目的は被写体の面積を求めたり、ラベリング(番号付け)、収縮、膨張、細線化などを行う前処理です。

#### ● いろいろな画像処理技術

写真5(a)は、花の部分の面積を求めるためにディジタイザでトレースしています。写真5(b)は、その面積の結果をパソコンのディスプレイに表示したものの例です。

2値化する場合は、原画像のうち目的とする像を背景から分離するための、スレッショルドまたはしきい値と呼ぶ濃度値を決める必要があります。しきい値は

画像を目で見ただけでは決めにくいいため、ヒストグラムで画像濃度の分布状態を求めると決定しやすい場合があります。

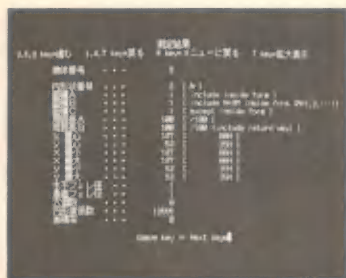
ヒストグラムは画像のなかの濃度値を、各濃度値ごとの個数としてグラフにしたものです。

写真6(a)は求めようとするエリアの所にカーソルを表示し、カーソルの内側のヒストグラムを求めます。

写真6(b)にヒストグラムを求めた例を示します。この写真に示すヒストグラムでは、背景と目的とする像にヒストグラムの谷の部分があるため、しきい値が求めやすくなっています。このヒストグラムだと、谷の部分にしきい値を設定します。

写真7は2値化像からラベリング、面積、重心などを求めたものをPC9801のディスプレイに表示したものです。

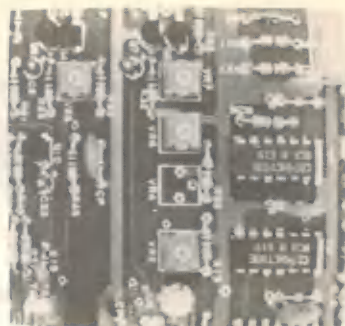




〈写真7〉 ラベリングした物体の結果表示



〈写真10〉 2×2ドット表示



〈写真13〉 実装忘れ基板  
(中央のポリウムがない)



〈写真8〉 ラベリングした物体を  
カーソルで指示する  
(カーソルは実際は点滅してる)



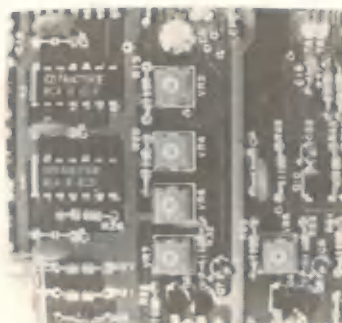
〈写真11〉 文字フォントの作成



〈写真14〉 差の表示



〈写真9〉 エッジの抽出結果  
(中心部のみ)



〈写真12〉 正常基板

また、テレビ・モニタの画像上のラベリングしたエリアにカーソルでわく取りをしています。その様子を写真8に示します。写真では見えにくいかもしれませんが、重心にもカーソルのポイントが出ています。

写真9は2値化した画像にエッジを抽出するフィルタをかけて、2値化像の輪郭を取り出したものです。わかりやすくするため、画像の中心部のみを処理してあります。

写真10は、画像データ濃度をPC9801のディスプレイに2×2ドットを1画素に対応させて、2×2ドットのなかの光らせるドット数を変化させ、5階調の画像に変換して表示したものです。この方法を使うとPC9801のグラフィック画面に画像を簡単に表示できます。

## ● 実際に使われている画像処理

実際にこの画像処理システムを応用した例として、文字のフォントを作成するシステムがあります。文字を手書きで紙に書き、それをテレビ・カメラで画像メモリに取り込み、PC9801の画面で修正して文字のフォント・データとして利用しています。

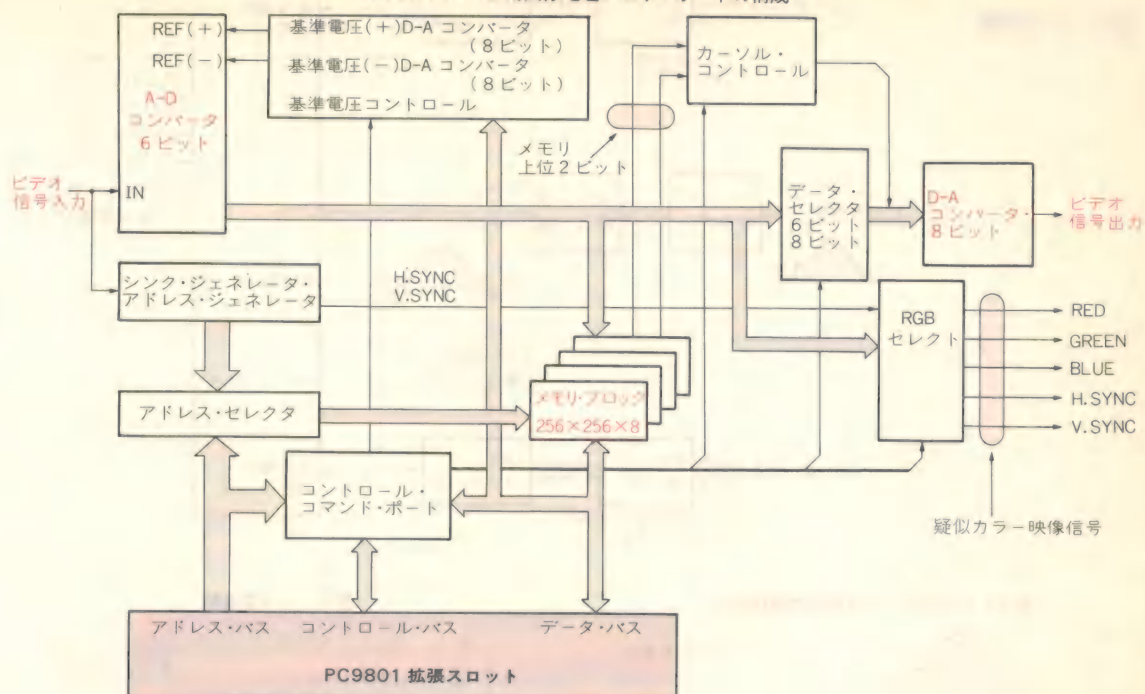
修正している様子の画面を、写真11に示します。この方法だと、文字フォントの作成が手書きに比べて修正が簡単なため、スピード・アップが図れます。

このほかに、2枚の画像を取り込み、その差を調べて画像間の違っている所だけを検出する用途があります。この応用では、プリント基板の部品の実装忘れなどが簡単に検出できます。

写真12は正常な基板、写真13は部品実装ミスのある



〈図1〉 パソコン用画像処理メモリ・ボードの構成



基板です。写真14はその差を表示したものです。この方法を使うとプリント基板の実装不良検査に応用できます。

## 画像処理システムのハードウェア

前述の実験に使った画像処理ボードについてハードウェアの説明をします。

このボードはPC9801の拡張スロットにダイレクトに挿入するだけで、あとはテレビ・カメラとモニター・テレビ、パソコン(PC9801)を用意すれば画像処理システムが構成できるものです。

図1にボードのブロック図を示します。このボードは大きく分けて四つのブロックになります。

第1は入力ビデオ信号をA-D変換する部分と、処理結果をモニター・テレビに出力するD-A変換部分です。第2はA-D変換した画像データを蓄えるメモリ部分です。

第3は同期信号を発生する部分で、第4はPC9801とのインターフェース部分から構成されています。

### ● A-D変換部

図2でA-D変換部の動作を説明します。カメラから入ってきたビデオ信号(NTSCコンポジット映像信号)は、バッファ・アンプを経由しブランキング部分(ビデオ信号の黒の基準となる電位)をクランプしてA-Dコンバータに入力します。

A-Dコンバータは、外部から供給される(+)の基準電圧と(-)の基準電圧の差を、このボードの場合は6ビットのデジタル・データに変換します。

もしA-Dコンバータの入力端子に入れる電圧が、(+)の基準電圧より高い場合はデータはオーバフローします。しかし、A-Dコンバータの出力は6ビットのフル・スケールのオール“1”です。このことをクリッピングされたと言います。

また(-)の基準電圧より低い電圧をA-Dコンバータの入力端子に入れた場合は、出力データはオール“0”になります。

このボードは、PC9801から書き込めるポートを持っていて、このポートに(+)と(-)の二つのデータを書き込むと、書き込んだデータが8ビットのD-Aコンバータに接続され、各々A-Dコンバータの(+)と(-)の基準電圧としてA-Dコンバータにつながっています。

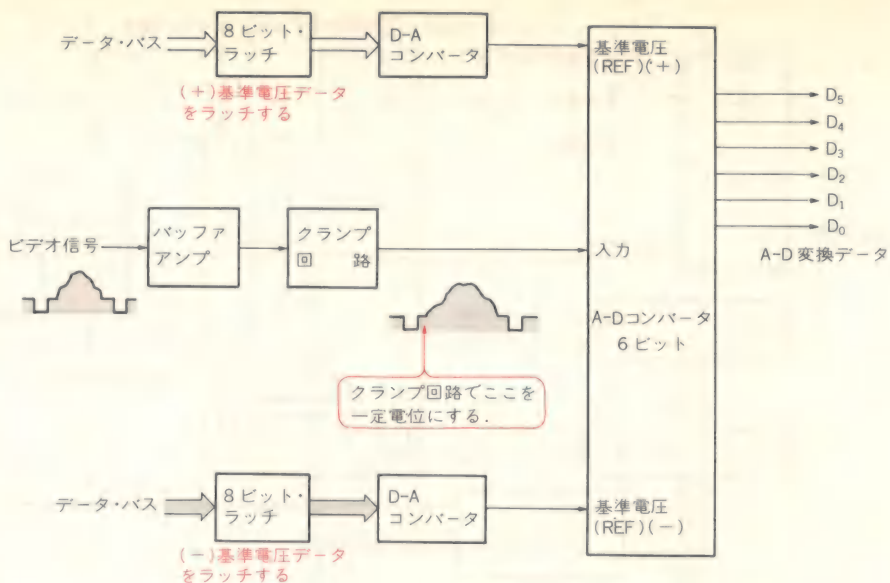
この各々の基準電圧を、入力の信号に応じて適宜に変化させることによって、任意のレベルの濃度域をA-D変換することができます。また、入力の映像信号のレベルが低くコントラストがない場合でも、(+)の基準電圧を入力レベルに応じて変化させると、A-Dコンバータの有効な取り込みができます。この説明を図3に示します。

### ● D-A変換部

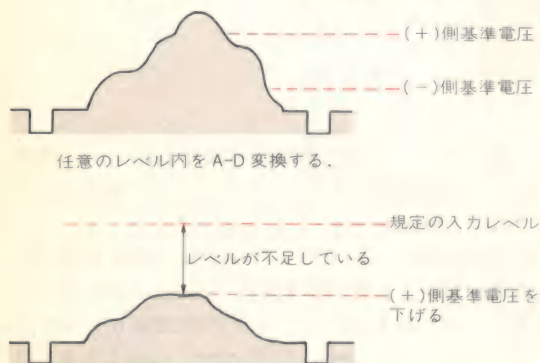
D-Aコンバータは図4のようになっており、メモリからのデータをアナログ信号に変換します。



〈図2〉 A-D変換部



〈図3〉 A-Dコンバータの基準電圧設定



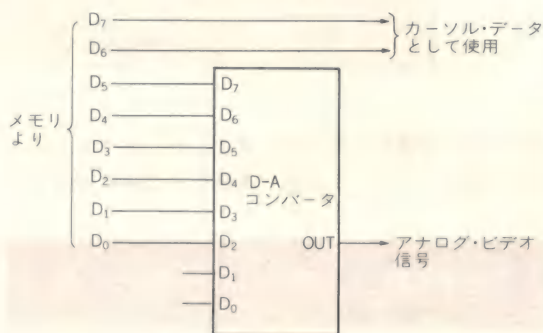
基準電圧をコントロールしてゲイン調整をする。

D-Aコンバータには8ビットのものを使っています。もしカーソルを使わないならば、メモリ・データの8ビットをそのままD-Aコンバータに接続することもできます。そして、PC9801からメモリに対して8ビットでデータを書き込むと、256階調の出力も行うことができます。

ここでは、メモリの上位2ビットはカーソル用として使います。2ビットありますので、1ビットずつ2枚のグラフィック・ビットとして画像信号に重畳します。写真15はこのカーソルを出した画像です。なお、カーソルの背景が明るい所に白のカーソルを書くと見えにくいので、カーソル部分の背景濃度が中間濃度(64階調の32階調目)より明るいとカーソルは黒になり、中間濃度より暗いと白くなるようになっています。

この方法を図5に示します。カーソルのビットと画像データのビット5のEX-ORをとり、データ・セクタの出力で画像データを強制的に白または黒に置き換

〈図4〉 D-A変換部



えます。こうすることで、背景の濃度によってカーソルが見えにくいということをなくしています。

### ● 画像メモリ

メモリ部分は64Kビット×4構成の256Kビットのダイナミック・メモリを8個使っています。

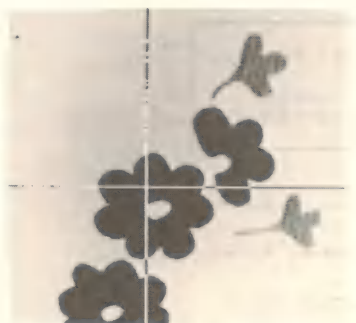
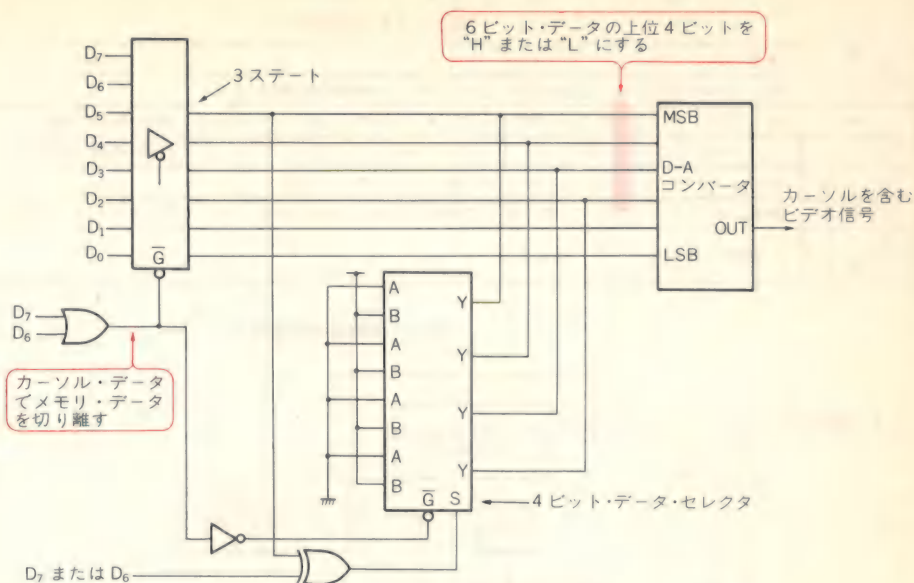
表1に示したように、1画面は縦横256画素で構成していますので、アドレスは65536となり、4画面分の容量がボード上に実装されています。また、メモリは図6で示すようにPC9801のメモリ空間の1バンクを占有します。ビデオ信号の取り込みはリアルタイムに取り込まれます。1バンクに重なって四つの画像バンクがあり、バンク・セレクトして画面の切り替えを行います。

このボードには、PC9801から書き込めるコマンド・ポートがあり、表2のようになっています。このポートのビット7を“H”から“L”に変化させると、メモリに対してライト信号が出されます。

このライト信号は、このポートで4画面書き込みモードに設定してあると、(4/60)秒間メモリに対してラ

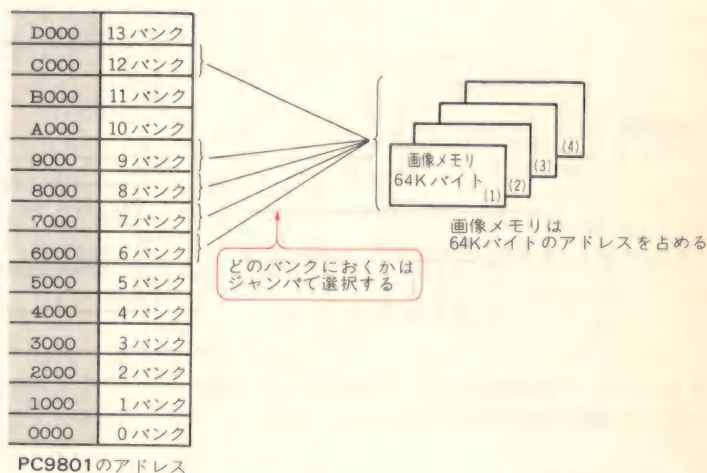


〈図5〉  
カーソルの表示方法



〈写真15〉 カーソル表示  
(背後の濃度により白黒反転する)

〈図6〉 画像メモリのマッピング



イト信号がでます。

メモリ素子のアドレスは、PC9801からリードまたはライトするときと、モニタ・テレビに表示しているときと両方からのアドレスを必要とするためアドレス切り替えをしています。

通常は表示するためのアドレスが供給されていますが、PC9801からメモリ・リードまたはライトのときのみ、PC9801のスロットに出るアドレスがメモリ素子に供給されるよう切り替えます。メモリ・アクセスはサイクル・スチールで行われているため、モニタ・テレビに画像を表示中に、PC9801からリード/ライトしても表示画面は乱れないようになっています。

サイクル・スチールの方法は、1走査線中を64分割し、交互にCPUアクセス時間と表示アクセス時間とに割り当てます。表示アクセス時間中にCPUアクセ

スが発生すると、CPUアクセス時間になるまでCPUにウェイトが発生します。

このCPUアクセス時間を作り出すために、リアルタイムにA-D変換しているデジタル・データを4画素分シリアル-パラレル変換し、4画素分を1アドレスとしてメモリに書き込みます。また、CPUより1画素単位でアクセスするには、PC9801のアドレス・バスの下位2ビットAB001, AB011を4画素の選択に用いて、AB021からAB151でメモリ素子のアドレスを構成しています。

PC9801から見たアドレスと画素は対応していますので、座標の変換は不要です。

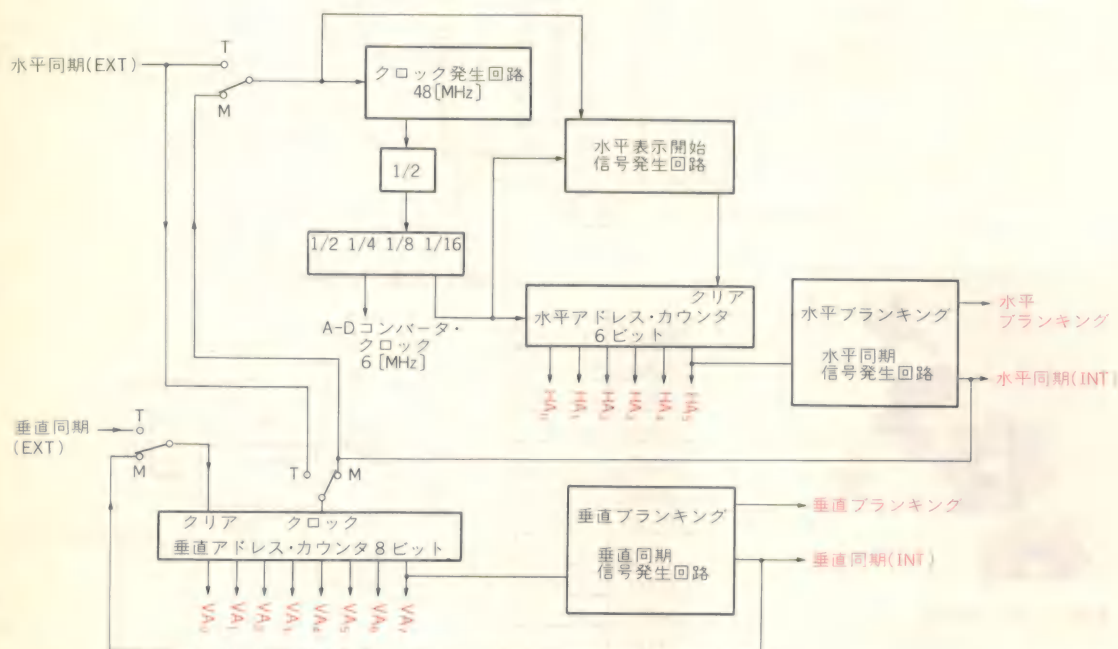
ダイナミック・メモリにつきもののリフレッシュは、常時画像出力を行うために読み出していますので自動的に行われます。テレビ信号の1走査線を64分割して



〈表2〉 コマンド・ポート

ビット 7	書き込みトリガ
ビット 6	取り込み画面数設定 (1画面取り込み、または4画面連続取り込み)
ビット 5	カーソル・コントロール (メモリ・データ・ビット7をカーソルとして有効/無効の選択)
ビット 4	カーソル・コントロール (メモリ・データ・ビット6をカーソルとして有効/無効の選択)
ビット 3	D-Aコンバータ、8ビット・モード/6ビット・モード選択
ビット 2	画像メモリのデータをディスプレイに出力するか、入力信号をスルーでディスプレイに出力するかを選択
ビット 1	画像メモリ・バンク・セレクト
ビット 0	

〈図7〉 同期信号発生部



いますので、1走査線でアドレスは64アドレス進み、4本の走査線で256アドレスとなってリフレッシュできます。

## ● 同期信号の作成

同期信号の作成はアドレス・カウンタと共用しています。

図7で示すように、入力信号から分離した、**水平および垂直の同期信号(EXT)**でアドレスのカウンタを**クリア**します。図7で示すスイッチは、入力画像をスルーでモニタ・テレビに表示するか、メモリの内容を読み出すかを選択します。この図ではメモリの読み出し側になっています。

水平同期信号は、水平同期(EXTまたはINT)でクロック発生回路のクロック・ゼネレータをクリアし、原発振の48MHzをイニシャライズします。

これを分周して、A-Dコンバータの変換クロックとして用いる6MHzと、水平アドレス・カウンタのクロックとなる1.5MHzを作ります。

水平アドレス・カウンタは、水平表示開始時間まで

クリアがかかっています。このクリアが解除されると、水平アドレスをカウントしていきます。

**水平アドレスは6ビットあり**、水平アドレスの最上位アドレスHA<sub>5</sub>が立ち下るとブランキング、水平同期パルス発生回路でブランキングおよび同期パルスを作ります。

水平画素数は256個なので、水平アドレスは8ビット必要ですが、図の分周器の1/8、1/16出力で残りの4画素を選択します。

垂直のアドレスは垂直同期(EXTまたはINT)で垂直アドレス・カウンタをクリアし、水平同期(EXTまたはINT)をクロックとしてカウントします。**垂直は256本ありますので8ビット必要**です。水平のときと同じように、最上位アドレスVA<sub>7</sub>が立ち下るとブランキング、垂直同期パルス発生回路でブランキングおよび同期パルスを作ります。

このように、アドレス・カウンタと同期信号発生器を一部兼用しています。

スイッチは、**入力画像を取り込むときと入力画像を**



〈表3〉疑似カラー、基準電圧  
コントロール・ポート

ビット 7 使用していない	ビット 7	基準電圧データ 8 ビット
ビット 6 } 疑似カラー・モード選択	ビット 6	
ビット 5 }	ビット 5	
ビット 4 }	ビット 4	
ビット 3 使用していない	ビット 3	
ビット 2 使用していない	ビット 2	
ビット 1 基準電圧のデータがプラスかマイナス選択	ビット 1	
ビット 0 基準電圧コントロールの有効/無効選択	ビット 0	

(a)

(b)

モニタリングするときはT側になり、メモリの内容を  
読むときはM側になります。画像を取り込んだ後でM  
側にしておけば、入力信号がなくなっても表示ができ  
ます。

PC9801の拡張バスから見たこのボードは、拡張メ  
モリになっています。このほか、ボードの動作  
を決めるためのポートが三つあります。三つのうち一  
つは表2です。そして、残り二つは表3に示します。  
表3で示したポートは、A-Dコンバータの基準電圧  
コントロールの有効/無効の切り替えと、次に説明す  
る疑似カラーの切り替え、および基準電圧のデータを  
書き込むポートです。

#### ● 疑似カラー出力部

最後に疑似カラーの出力部の説明をします。疑似カ  
ラーとは、取り込んだ6ビットのデータの各ビットを  
デジタルのRED(赤)、GREEN(緑)、BLUE(青)の  
各色に表4に示すように割り当てます。

また、RGBの割当は4種類に変更できます。変更  
するには、表3(a)のポートの疑似カラー・モードを書  
き換えます。表4の(a)のときは、6ビットの濃度が明  
るい順に白、黄、水色、緑、紫、赤、青、黒、と割り  
当てられます。表4の(b)、(c)、(d)と切り替えていくと、  
(a)の時の1色の中がより細かく表現できます。これを  
巻頭の口絵ページの写真に示します。この方法によっ  
て、白黒画面では判別しにくい濃度変化を、よりはっ  
きりととらえることができます。

## 画像処理プログラムの設計法

今回使ったシステムで、簡単な画像処理プログラム  
を作ってみました。環境は以下のとおりです。

パソコン PC9801VX2

ハード・ディスク 20Mバイト

RAMディスク 2Mバイト

MS-DOS Ver2.11

MASM Ver3.0

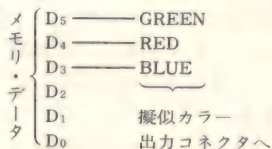
LINK Ver3.0

Microstft C Ver3.0017

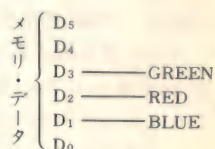
処理の内容は以下のように決めました。

- (1) カメラからの画像をメモリに取り込む。
- (2) 取り込んだ画像の白黒を反転させる。

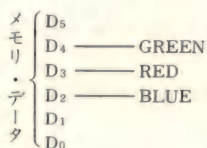
〈表4〉疑似カラー出力



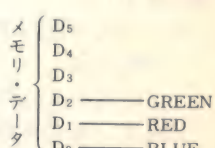
(a)



(c)



(b)



(d)

- (3) 反転させた画像を2値化する。

以上の処理を行うために、Microsoft C(以下MS-  
C)で使える関数をアセンブラで作成しました。ここ  
で作成したプログラムはすべてスモール・モデル用で  
す。

アセンブラのプログラムをリスト1に示します。

#### ● 基本的な使い方

ここで使用した関数は `f d m g e t 1 4 ( )`、`f p  
e e k ( )`、`f p o k e ( )` です。

`f d m g e t 1 4` は画像メモリのコマンド・ポート  
を4画面モードにし、書き込みトリガを“H”から  
“L”に変化させます。これで画像メモリは、入力信  
号を4画面分リアルタイムでメモリに取り込みます。  
実際は、コマンド・ポートのほかの状態を変化させな  
いで書き込みトリガだけを変える必要がありますが省  
略します。

`f p e e k` は画像メモリのセグメント・アドレスと  
X、Y座標を指示して、画像メモリのデータを読み出  
します。

`f p o k e` は画像メモリのセグメント・アドレスと  
X、Y座標、書き込むデータを指示して画像メモリに  
書き込みます。

座標は画面の横方向をX座標、縦方向をY座標とし  
ます。

以上の関数を使い、最初に決めた処理を行います。  
そのプログラムをリスト2に示します。リスト1とリ



スト2のアセンブルとコンパイル、リンクはリスト3のバッチ・ファイルで行います。

画像処理のプログラムは、このようにX、Y座標を移動しながら画素データを読み込み、そのデータを加工して元の座標にもどすという処理が多くあります。

### ● 自動2値化の方法

このプログラムで行った2値化は、2値化する濃度が一定でしたが、次に**自動2値化のアルゴリズム**を簡単に説明します(この方法は大阪大学工学部応用物理学教室 岡研究室のご協力による)。

方法は図8(a)に示したように、 $9 \times 9$  エリアの濃度の平均値をしきい値として、 $9 \times 9$  の中心の  $A_5$  の  $3 \times 3$  のエリアを2値化します。

まず最初に  $A_1 - A_9$  のそれぞれのエリアの平均値を求めます。

$$A_1 = (P_{11} + P_{12} + P_{13} + P_{21} + P_{22} + P_{23} + P_{31} + P_{32} + P_{33}) / 9$$

エリアの  $A_2$  から  $A_9$  についても同様に求めます。

次に  $A_1 - A_9$  の平均値  $A$  を求めます。

$$A = (A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8 + A_9) / 9$$

ここで求められた  $A$  をしきい値として、 $A_5$  の  $3 \times 3$  のエリアを図(b)に示すように2値化します。

画像の濃度階調を0～63の64階調として、2値化する画素の濃度を  $P$ 、しきい値を  $A$  とすると、

$$A \leq P \text{ のときは } P = 63 \text{ (白)}$$

$$A > P \text{ のときは } P = 0 \text{ (黒)}$$

とすることにより2値化像が得られます。

例えばある  $3 \times 3$  のエリアが図9(a)であるとき、しきい値が32であれば、結果は図(b)のようになります。この操作を1画面全体について行いますが、X座標を3画素ずつ移動していき、Y座標についても同様に行います。

この方法による自動2値化では、画面の端の3画素については2値化できません。また3で割り切れないものの端数のエリアについても2値化できないため、

これらのエリアを“0”とする必要があります。

2値化にはこのほかいろいろの方法が考えられますが、希望するしきい値を決定することは難しいようです。

### ● コントラストを強調するには

次に入力した画像のレベルが小さい場合、次のような処理を行うとコントラストが強調されます。

まず、処理するエリア全体の最高濃度  $H_a$ 、最低濃度  $H_i$  を求めます。処理する前の画素の濃度を  $L_P$  とし、処理後の濃度を  $L_X$  とします。そして各画素ごとに次の計算を行います。

$$L_X = 63 \times (L_P - H_i) / (H_a - H_i)$$

この方法を使い処理した画像を写真16に示します。処理は画面の中心だけを行っています。

### ● マスク処理について

画像処理の方法では、このほかにマスク処理というものがあります。マスク処理は注目した1画素に隣接する8画素と、マスク・データとの演算結果を注目した1画素に反映させる処理です。

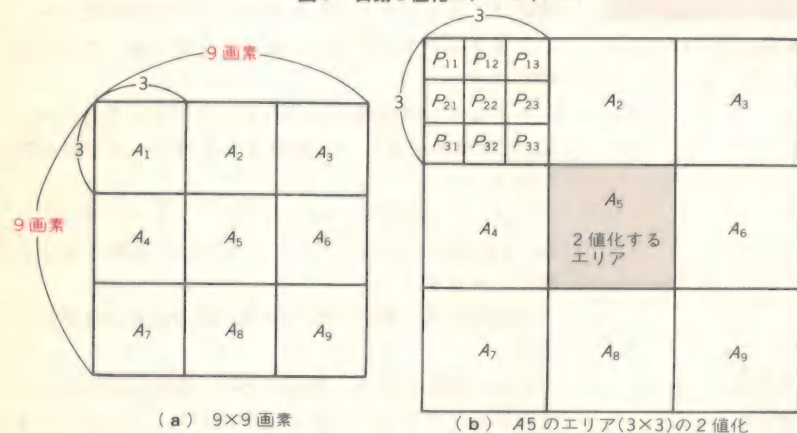
この方法で画面のノイズを減少させることができます。マスク・データとしては図10を使います。

演算の方法は、隣接する画像データとマスク・データを次式のように計算します。図11の  $P_0$  は、注目した画素の濃度、 $P_1$  から  $P_8$  は隣接した画素の濃度です。また  $M_1$  から  $M_8$  はマスク・データです。  $P_A$  は演算結果

〈写真16〉  
中央部のみ  
コントラスト強調



〈図8〉 自動2値化のアルゴリズム



〈図9〉 濃度階調の2値化





〈図10〉 ノイズ減少マスク・データ

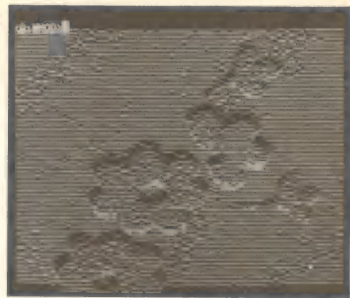
$M_1$	$M_2$	$M_3$
$M_4$	$M_0$	$M_5$
$M_6$	$M_7$	$M_8$

$M_0 \sim M_8$  はすべて1とする。  
また、 $M_0$  を8とする場合もある。

〈図11〉 注目画素に隣接した画素データ

$P_1$	$P_2$	$P_3$
$P_4$	$P_0$	$P_5$
$P_6$	$P_7$	$P_8$

$P_0$  : 注目画素



〈写真17〉 3次元表示

の濃度値です。

$$P_A = (P_1 \cdot M_1 + P_2 \cdot M_2 + P_3 \cdot M_3 + P_4 \cdot M_4 + P_5 \cdot M_5 + P_6 \cdot M_6 + P_7 \cdot M_7 + P_8 \cdot M_8 + P_0 \cdot M_0) / 9$$

このマスク・データの場合では、処理後の画像のぼけが大きいため、マスク・データの $M_0$ を8にした次式の方法もあります。

$$P_A = (\text{ここは同じ}) / 16$$

#### ● 濃度を3次元的に表示するには

最後に取り込んだ画像の濃度を読み、その濃度を3次元的にPC9801のディスプレイに表示するプログラムを紹介します。

このプログラムでは、先のリスト1を拡張して、MS-Cで使えるグラフィック・ライブラリ**SLIB98.LIB**(自作)と、この画像ボード専用のライブラリ**SLIBFDM.LIB**(自作)を使用しています。MS-C用の市販されているグラフィック・ライブラリもあります。

使用した関数を次に示します。

**ginit()** PC9801のグラフィックLIOを初期化します。PC9801のグラフィックを使う場合は最初に必ずこの関数を呼びます。

**gcls()** PC9801のグラフィック画面をクリアします。

**tcls()** PC9801のテキスト画面をクリアし

ます。

**tllocate(x, y)** PC9801のカーソル位置をx, yに移動します。

**fdmunit(FDM-SEG, WORK-SEG, FDMPORT-A, FDMPORT-B, FDMPORT-C)**

画像ボードの画像メモリ・アドレス、ポート・アドレス、ワーク・アドレスを初期化します。

**gline(x0, y0, x1, y1, palette, style)**

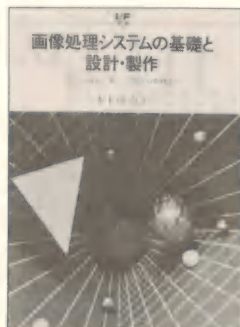
PC9801のグラフィック画面に、palette, styleで指定したラインを書きます。

リスト4にMS-Cで書いたプログラムを示します。リスト5にリスト4のプログラムのコンパイル、リンクを行うバッチ・ファイルを示します。

写真17にその結果を示します。

#### ●参考文献●

- (1) 長尾真；デジタル画像処理，(株)近代科学社，昭和60年5月1日，第8刷。
- (2) 吹抜敬彦；FAX，OAのための画像の信号処理，日刊工業新聞社，昭和57年10月20日，初版第1刷。
- (3) 田村秀行；コンピュータ画像処理入門，総研出版，昭和60年8月10日，第1版。



I/F BOOKS

好評発売中

## 画像処理システムの基礎と設計・製作

Z80システムで学ぶ入門から実践まで

木下健治 著

A5判 定価 1,545円

216頁

マイクロコンピュータの及ぼした影響のひとつに、パターン処理技術の進歩があり、その典型的なものとして画像処理技術が普及しつつあります。本書は、パソコンを効果的に使い、身近に画像処理システムを構築しようという目的で書かれたものです。ハード/ソフト共に、その実用技術を公開しています。

CQ出版社 ◇ ◇ ◇ 〒170 東京都豊島区巢鴨1-14-2



```

1: 1: *****
2: 2: For Microsoft C(Ver 3.0017)
3: 3:
4: 4: Small Model
5: 5: *****
6: 6:
7: 7: FDM98-4 TEST Program May 1987
8: 8: *****
9: 9: MACRO definition for Microsoft C Version 3.0
10: 10: *****
11: 11:
12: 12:
13: 13:
14: 14:
15: 15:
16: 16:
17: 17:
18: 18:
19: 19:
20: 20:
21: 21:
22: 22:
23: 23:
24: 24:
25: 25:
26: 26:
27: 27:
28: 28:
29: 29:
30: 30:
31: 31:
32: 32:
33: 33:
34: 34:
35: 35:
36: 36:
37: 37:
38: 38:
39: 39:
40: 40:
41: 41:
42: 42:
43: 43:
44: 44:
45: 45:
46: 46:
47: 47:
48: 48:
49: 49:
50: 50:
51: 51:
52: 52:
53: 53:
54: 54:
55: 55:
56: 56:
57: 57:
58: 58:
59: 59:
60: 60:
61: 61:
62: 62:
63: 63:
64: 64:
65: 65:
66: 66:
67: 67:
68: 68:
69: 69:
70: 70:
71: 71:
72: 72:
73: 73:
74: 74:
75: 75:
76: 76:
77: 77:
78: 78:
79: 79:
80: 80:
81: 81:
82: 82:
83: 83:
84: 84:
85: 85:
86: 86:
87: 87:
88: 88:
89: 89:
90: 90:
91: 91:
92: 92:
93: 93:
94: 94:
95: 95:
96: 96:
97: 97:
98: 98:
99: 99:
100: 100:
101: 101:
102: 102:
103: 103:
104: 104:
105: 105:
106: 106:
107: 107:
108: 108:
109: 109:
110: 110:
111: 111:
112: 112:
113: 113:
114: 114:
115: 115:
116: 116:
117: 117:
118: 118:
119: 119:
120: 120:
121: 121:
122: 122:
123: 123:
124: 124:
125: 125:
126: 126:
127: 127:
128: 128:
129: 129:
130: 130:
131: 131:
132: 132:
133: 133:
134: 134:
135: 135:
136: 136:
137: 137:
138: 138:
139: 139:
140: 140:
141: 141:
142: 142:
143: 143:
144: 144:
145: 145:
146: 146:
147: 147:
148: 148:
149: 149:
150: 150:
151: 151:
152: 152:
153: 153:
154: 154:
155: 155:
156: 156:
157: 157:
158: 158:
159: 159:
160: 160:
161: 161:
162: 162:
163: 163:
164: 164:
165: 165:
166: 166:
167: 167:
168: 168:
169: 169:
170: 170:
171: 171:
172: 172:
173: 173:
174: 174:
175: 175:
176: 176:
177: 177:
178: 178:
179: 179:
180: 180:
181: 181:
182: 182:
183: 183:
184: 184:
185: 185:
186: 186:
187: 187:
188: 188:
189: 189:
190: 190:
191: 191:
192: 192:
193: 193:
194: 194:
195: 195:
196: 196:
197: 197:
198: 198:
199: 199:
200: 200:
201: 201:
202: 202:
203: 203:
204: 204:
205: 205:
206: 206:
207: 207:
208: 208:
209: 209:
210: 210:
211: 211:
212: 212:
213: 213:
214: 214:
215: 215:
216: 216:
217: 217:
218: 218:
219: 219:
220: 220:
221: 221:
222: 222:
223: 223:
224: 224:
225: 225:
226: 226:
227: 227:
228: 228:
229: 229:
230: 230:
231: 231:
232: 232:
233: 233:
234: 234:
235: 235:
236: 236:
237: 237:
238: 238:
239: 239:
240: 240:
241: 241:
242: 242:
243: 243:
244: 244:
245: 245:
246: 246:
247: 247:
248: 248:
249: 249:
250: 250:
251: 251:
252: 252:
253: 253:
254: 254:
255: 255:
256: 256:
257: 257:
258: 258:
259: 259:
260: 260:
261: 261:
262: 262:
263: 263:
264: 264:
265: 265:
266: 266:
267: 267:
268: 268:
269: 269:
270: 270:
271: 271:
272: 272:
273: 273:
274: 274:
275: 275:
276: 276:
277: 277:
278: 278:
279: 279:
280: 280:
281: 281:
282: 282:
283: 283:
284: 284:
285: 285:
286: 286:
287: 287:
288: 288:
289: 289:
290: 290:
291: 291:
292: 292:
293: 293:
294: 294:
295: 295:
296: 296:
297: 297:
298: 298:
299: 299:
300: 300:
301: 301:
302: 302:
303: 303:
304: 304:
305: 305:
306: 306:
307: 307:
308: 308:
309: 309:
310: 310:
311: 311:
312: 312:
313: 313:
314: 314:
315: 315:
316: 316:
317: 317:
318: 318:
319: 319:
320: 320:
321: 321:
322: 322:
323: 323:
324: 324:
325: 325:
326: 326:
327: 327:
328: 328:
329: 329:
330: 330:
331: 331:
332: 332:
333: 333:
334: 334:
335: 335:
336: 336:
337: 337:
338: 338:
339: 339:
340: 340:
341: 341:
342: 342:
343: 343:
344: 344:
345: 345:
346: 346:
347: 347:
348: 348:
349: 349:
350: 350:
351: 351:
352: 352:
353: 353:
354: 354:
355: 355:
356: 356:
357: 357:
358: 358:
359: 359:
360: 360:
361: 361:
362: 362:
363: 363:
364: 364:
365: 365:
366: 366:
367: 367:
368: 368:
369: 369:
370: 370:
371: 371:
372: 372:
373: 373:
374: 374:
375: 375:
376: 376:
377: 377:
378: 378:
379: 379:
380: 380:
381: 381:
382: 382:
383: 383:
384: 384:
385: 385:
386: 386:
387: 387:
388: 388:
389: 389:
390: 390:
391: 391:
392: 392:
393: 393:
394: 394:
395: 395:
396: 396:
397: 397:
398: 398:
399: 399:
400: 400:
401: 401:
402: 402:
403: 403:
404: 404:
405: 405:
406: 406:
407: 407:
408: 408:
409: 409:
410: 410:
411: 411:
412: 412:
413: 413:
414: 414:
415: 415:
416: 416:
417: 417:
418: 418:
419: 419:
420: 420:
421: 421:
422: 422:
423: 423:
424: 424:
425: 425:
426: 426:
427: 
```



### <リスト2>

```

1: #include
2: #define FDM_SEG 0xC000 /* f d m セグメント
3: /*
4: #define MASK_DATA 0x3f /* マスクデータ
5: /*
6: #define MAX_DATA 63 /* 6ビット最大データ
7: /*
8: #define MIN_DATA 0 /* 6ビット最小データ
9: /*
10: main()
11: {
12: fdmset14();
13: neg();
14: twof();
15: FDM_SCREEN NEGATE
16: -----*/
17: neg()
18: {
19: int dat;
20: int i, j;
21: for (i = 0 ; i <= 255 ; i++)
22: {
23: for (j = 15 ; j <= 255 ; j++)
24: {
25: dat = fpeek(FDM_SEG, i, j) & MASK_DATA;
26: dat = ~ dat;
27: fpoke(FDM_SEG, i, j, dat);
28: }
29: }
30: }
31: }
32: /*
33: データ2補正 (3 f h のマスク)
34: -----*/
35: twof()
36: {
37: int dat;
38: int i, j;
39: int thresh = 31;
40: for (i = 0 ; i <= 255 ; i++)
41: {
42: for (j = 15 ; j <= 255 ; j++)
43: {
44: dat = fpeek(FDM_SEG, i, j) & MASK_DATA;
45: if (dat > thresh) dat = MAX_DATA;
46: else
47: dat = MIN_DATA;
48: fpoke(FDM_SEG, i, j, dat);
49: }
50: }
51: }
52: }

```

### <リスト3>

```

1: MASM MSC, MSC;
2: MSC SAMPLE1 /IA: * INCLUDE Y;
3: LINK SAMPLE1 MSC, SAMPLE1;

```

### <リスト4>

```

1: #define FDM_SEG 0xC000 /* Gazou memory segment
2: #define WORKSEG 0x8000 /* work segment
3: #define FDMPORT_A 0x00D0 /* Command port
4: #define FDMPORT_B 0x00D4
5: #define FDMPORT_C 0x00D8
6: #define FDMPORT_D 0x00DC
7: #define fmax 255
8:
9: main()
10: {
11:
12: gclis(3, 7); tlocate(23, 0); tcis();
13: fdmainit(FDM_SEG, WORKSEG, FDMPORT_A, FDMPORT_B, FDMPORT_C);
14: three();
15:
16: }
17:
18: three()
19: {
20: int fy0, fy, x0, x1, x, y, lx0=11, ly0=20, fyoff=20;
21:
22: for (y = ly0; y < 256; y += 3)
23: {
24: x0 = 0;
25: fy0 = fyoff-(fpeek(FDM_SEG, lx0-2, y));
26: for (x = lx0; x < fmax; x += 2)
27: {
28: xl = ((x-lx0)*5)/3;
29: fy = fyoff-(fpeek(FDM_SEG, x, y)/5);
30: gline(x0, fy0, x1, fy, 4, -1);
31: fy0 = fy; x0 = x1;
32: }
33: fyoff += 2;
34: }

```

### <リスト5>

```

1: MSC SAMPLE2 /IA: * INCLUDE Y;
2: LINK SAMPLE2, SAMPLE2, nul, A: * LIB XSLIBC * SLIB98 * SLIBFDM

```



## パソコン用画像入力ボードの 設計・製作



小野隆夫

パソコン用画像入力ボードも、高機能を望まなければ、低価格に実現できます。ここでは特別なA-Dコンバータや画像メモリを必要としない、多階調の画像入力ボードの実例を紹介します。

ここ数年のマイコンの処理能力の向上は目ざましく、10年前には考えられなかった個人レベルでの画像処理が、現在では十分可能な状態になってきました。

画像入力装置には、ディジタイザ、タブレット、ライト・ペンといった従来からの装置のほかに、最近ではイメージ・スキャナといったハンド・スキャン方式の簡易画像入力装置も製品化されるようになりました。

ところで、上記のものよりコスト高にはなりますが、効率のよい入力装置として、テレビ・カメラを使用したものがあります。これは**フレーム・メモリなどとも呼ばれ、1/30秒ほどで画像全体を入力してしまいます**。つまり、高速A-Dコンバータによって量子化された画像データを、CPUを介さずに次々と画像メモリに書き込むという、DMA方式で動作しているものです。

しかし、この方式は、大容量の画像メモリと高速A-Dコンバータが必要とされます。そこで、フレーム・メモリの1/10～1/100のコストで、テレビ・カメラからの画像入力ができないかと考えたのが、ここで述べるカメラ・インターフェース・システムです。

ここで製作するインターフェースには、次のような特徴があります。

- (1) NTSC方式に準拠する白黒テレビ・カメラが接続される
- (2) **静止画を2値化して入力する**
- (3) **8階調入力も可能**
- (4) ホスト・コンピュータとしてPC9801を使う
- (5) PC9801の後部スロットに納まるサイズ
- (6) フレーム・メモリの1/10～1/100の製作コスト
- (7) 1画面の入力時間は約1.4秒。

### 画像入力インターフェースの構成

ビデオ・カメラの画像を入力する装置として、画像メモリ装置(フレーム・メモリ装置)というのがあり、数種類のものが市販されています。これらは、映像信号をサンプリングして、明るさに応じてディジタル化(量子化)した画像データを、DMAにより次々とリフレッシュ・メモリに入力します。1画面につき、1/30

秒程度で入力できますが、高速A-Dコンバータと大容量のリフレッシュ・メモリを必要とします。

表1が、DMA方式とプログラム方式の相違点です。入力に要する時間が多少かかってもよい、または2値化が入力できればよい、といった用途には、安価で小型に製作できるプログラム制御方式がよいでしょう。

#### ● 動作の概略

プログラム制御方式によって画像入力をするためのインターフェース構成は、図1に示したとおりです。

1フレームは図2のように座標化されており、データの欲しい座標をインターフェースに与え、サンプリング要求をすると、1/30秒以内にデータを用意して、レディ信号を返送します。これが動作の概略です。

走査線カウンタは、図3の走査線を数えます。ただし、上から順に1, 264, 2, 265...と数えるのではなく、これを**0, 1, 2, 3...と数えていきます**。

第1フィールドでは、1, 2, 3...263で、第2フィールドでは、263, 264...525の走査線は、コンピュータから見たときに、**第1フィールドで、0, 2, 4...となり、第2フィールドで1, 3, 5...となります**。したがって、1フレームは、カウンタ値が偶数になるフィールドと、奇数になるフィールドとから成り立っていると見なすことができます。

水平位置カウンタ値は、1本の走査線の左端からの大きさを示します。

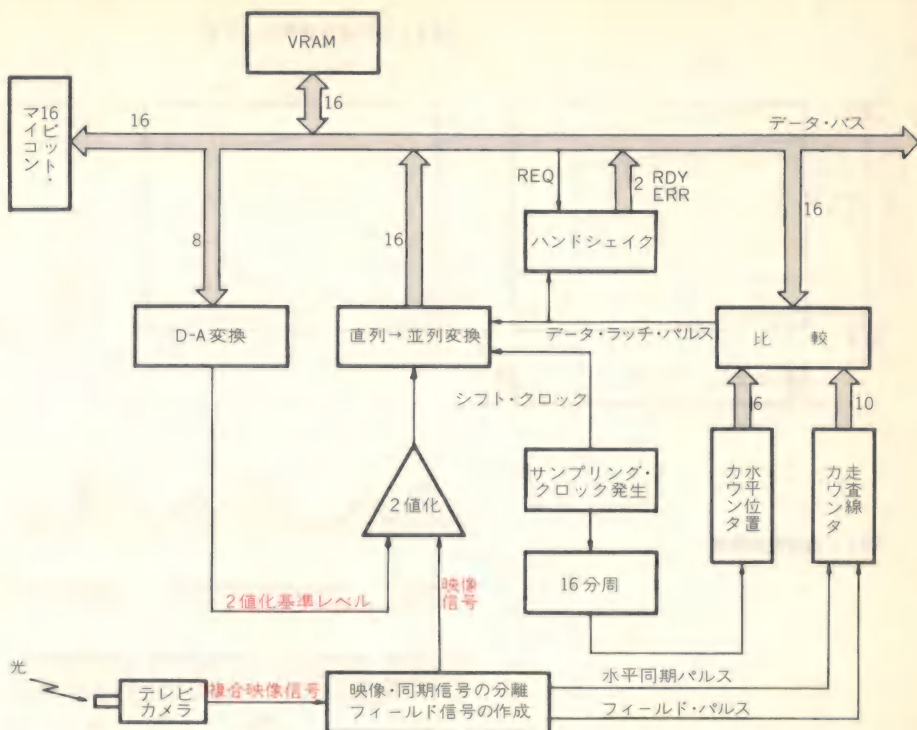
両カウンタによって、1フレームはVRAMの座標

〈表1〉 DMA方式とプログラム方式の相違点

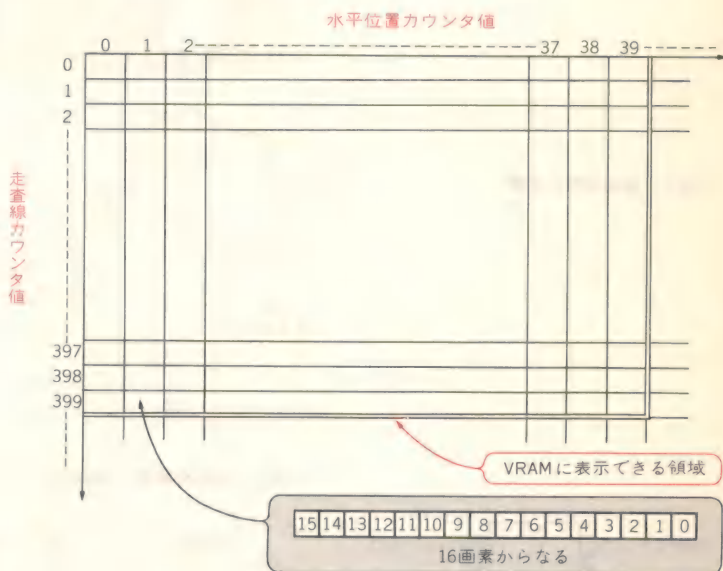
	DMA方式	プログラム方式
入力方法	ハードウェア制御(ダイレクト・メモリ・アクセス方式)	ソフトウェア制御(CPUによるアクセス方式)
入力時間	リアル・タイム	CPUの動作速度によるが、DMA方式の数十倍以上かかる
A-D変換器	高速なもの(一般的に200ns程度)が要求される	低速なものでもよい
画像メモリ	専用の外部メモリが必要	不要
コスト	高い	DMA方式の数分の1～数十分の1



〈図1〉  
カメラ・インターフェ-  
ース・システムの構成



〈図2〉 フレームの座標化



のように扱うことができます。図2の縦400ライン、横40ワードを取り出すと、それはちょうど640×400のVRAMに表示できる領域になります。

図1にある直列-並列変換の並列出力の16ビットには、走査線カウンタ、水平位置カウンタによってポイントされた、フレーム内のある位置の画像データを2値化したものが出力されています。このデータは、両カウンタ値の変化に伴い、次々に更新されます。

もし、両カウンタ値がコンピュータから比較回路に

与えられた座標値と一致したならばデータはラッチされ、ハンドシェイク・ポートからレディ信号が送出されます。

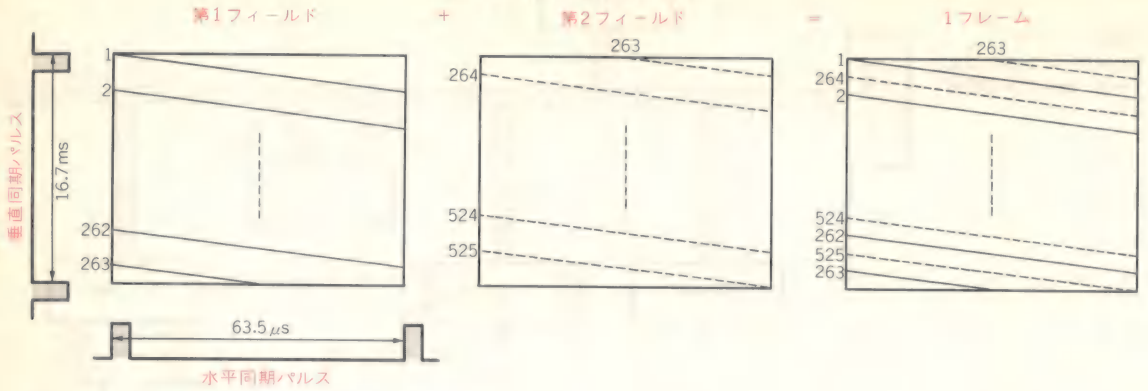
図1のD-A変換回路に与える値によって、2値化の比較レベルが変化します。これによって、8階調の表現や照度の自動補正などが、ソフトウェアによって可能になります。

#### ● 同期信号の分離

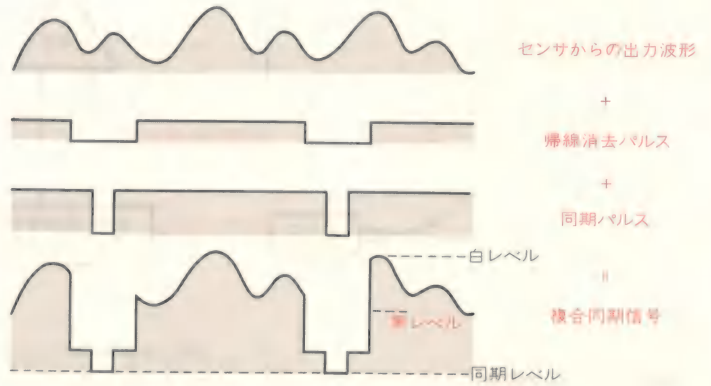
図4にあるように、テレビ・カメラの出力信号は、



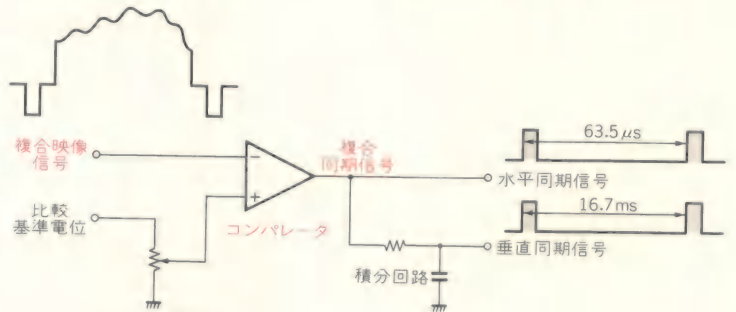
〈図3〉飛び越し走査のようす



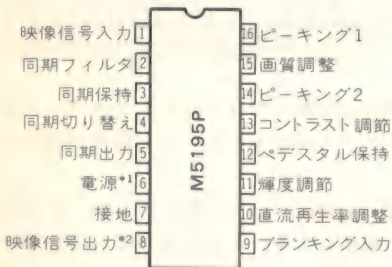
〈図4〉複合映像信号



〈図5〉同期信号の分離

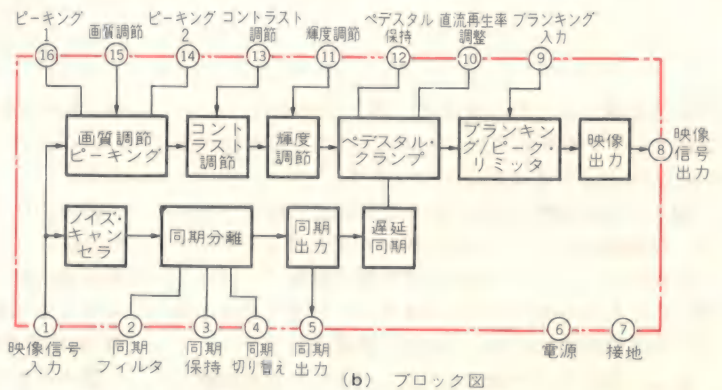


〈図6〉同期分離用IC M5195P(三菱)



- (※1) 定格電源電圧: 12.0V
- (※2) 同期出力パルス振幅: 標準 10.5V<sub>P-P</sub>

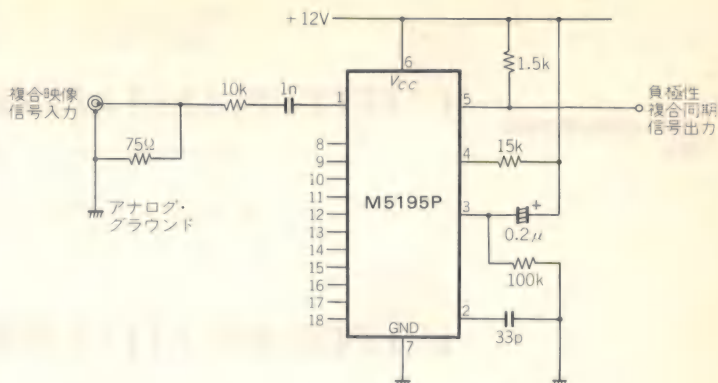
(a) 端子接続図



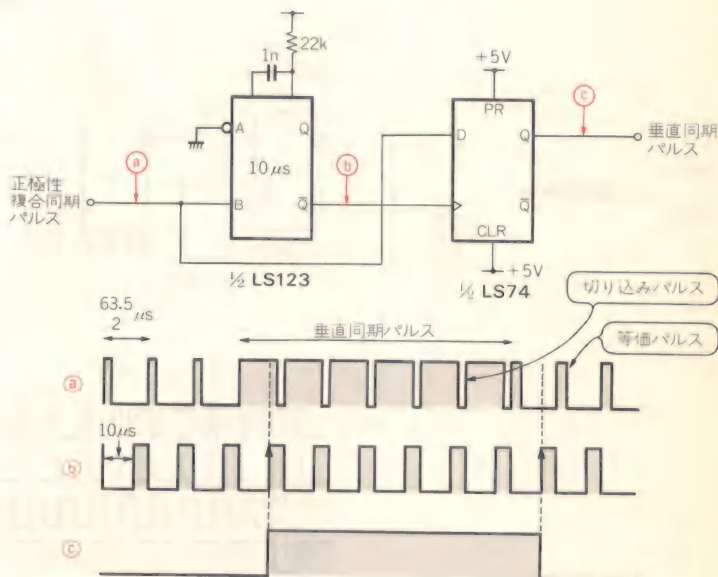
(b) ブロック図



〈図7〉同期分離回路



〈図8〉垂直同期信号の分離



複合映像信号です。図のように、映像信号の黒レベルと同期信号レベルの間にしきい値を設けたコンパレータによって、容易に分離できます。複合映像信号から分離した同期信号を、積分回路に通して垂直同期信号を得ます。図5に分離の概念図を示します。

ここでは、同期信号を分離するのに専用の同期分離ICを使い、垂直同期信号の分離はデジタル的にを行います。

図6に、同期分離用IC M5195Pのブロック図とピン接続図を示します。このICは、映像信号の処理ができますが、ここでは同期分離の部分しか使っていません。図7に回路例を示します。

図8が、複合同期信号から垂直同期信号を分離する回路です。

#### ● フィールド識別信号の作成

図3のように、1フレームは2フィールドからできているので、第1フィールドの終わり～第2フィールドの始めと、第2フィールドの終わり～第1フィールドの始めのようすは、図9のようになっています。

1フレームを正しく作るには、両フィールドの重ね方が問題となります。

第1フィールドと第2フィールドを識別するための信号をつくる回路を図10に示します。

ここでは、垂直同期信号の後ろ側に現れる等化パルスの数の違いを利用しています。図9のような正確な標準信号を作るのはめんどろなため、テレビ・カメラの中には、等化パルス、切り込みパルスが簡略化されているものがあります。数種類のカメラで試してみましたが、この回路でいずれも正しい論理のフィールド識別信号が得られました。

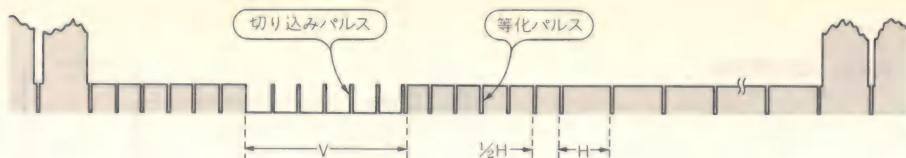
#### ● サンプリング回路の構成

図4のように、連続した映像信号はアナログ信号ですので、これを離散的なデジタル信号に変換しなければなりません。このためには、普通、図11にあるようなサンプリングと量子化を行います。

入力信号は、特別なスイッチであるサンブラを通ります。スイッチは、サンプリング・パルスによって時間間隔 $t$ で開閉されます。このスイッチの開閉が、い



〈図9〉 実際の複合映像信号



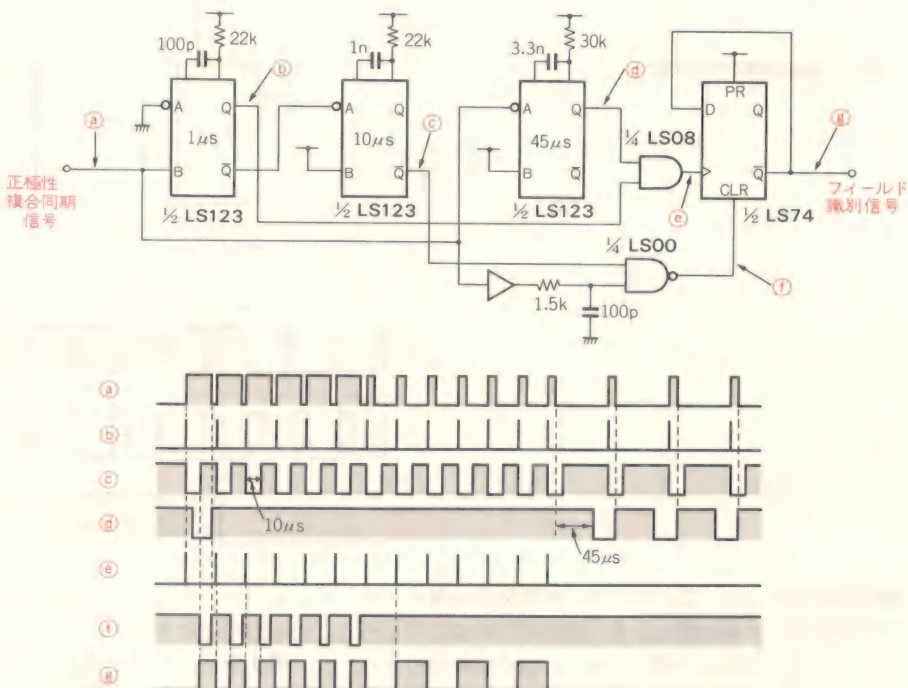
(a) 第2フィールドの終わり～第1フィールドの始め



(b) 第1フィールドの終わり～第2フィールドの始め

[V: 垂直同期期間  
H: 水平同期期間]

〈図10〉 フィールド識別信号の作成



わばサンプリングに当たります。

スイッチを閉じたときに、出力にはその時刻の入力信号の大きさが現れます。しかし、その値は無限の精度で得られるのではなく、例えば四捨五入して整数化されるわけです。一言でいえば、この整数化を量子化といい、四捨五入したことによる誤差を量子化誤差といいます。

走査線1本分を横640ドットのVRAMに表示するためには、どのくらいの周波数のサンプリング・パルスを与えればよいでしょうか。

水平走査の周期は $63.5\mu\text{s}$ ですが、この中には画像データではない水平同期パルスと水平帰線消去パルスの期間が含まれ、それは最小16.5%あります。したがって、画像表示に関係する期間は、 $53.0\mu\text{s}$ になります。この期間に640のサンプリング点を設ければ、サ

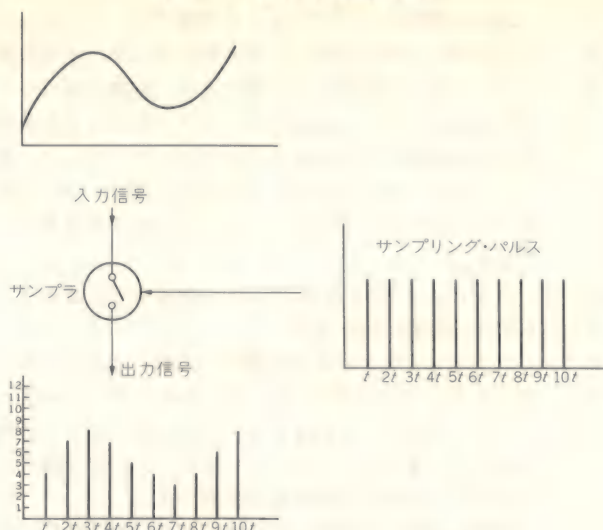
ンプリング周期は $82.8\text{ns}$ となり、サンプリング周波数は約 $12\text{MHz}$ となります。図12にあるのが、サンプリング・パルス発生回路で、周波数が可変で発振停止のコントロールができます。

さて、ここでは、映像信号を2値化するだけなので、量子化は容易です。映像信号をあらかじめ定めた基準レベルと比較し、直列-並列変換のために、シフトレジスタに入力します。シフトレジスタのシフト・クロックとして、サンプリング・パルスを加えます。するとシフトレジスタの並列出力には、デジタル化された画像データが出力されています。

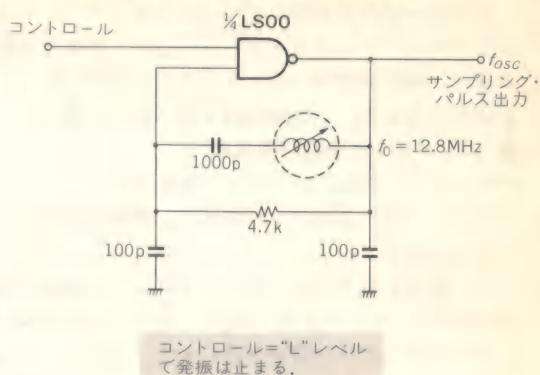
2値化のための基準電位は、D-Aコンバータによって与えます。コンピュータから、2値化レベルをソフトウェアによって変更できるようにするためです。この回路を図13に示します。



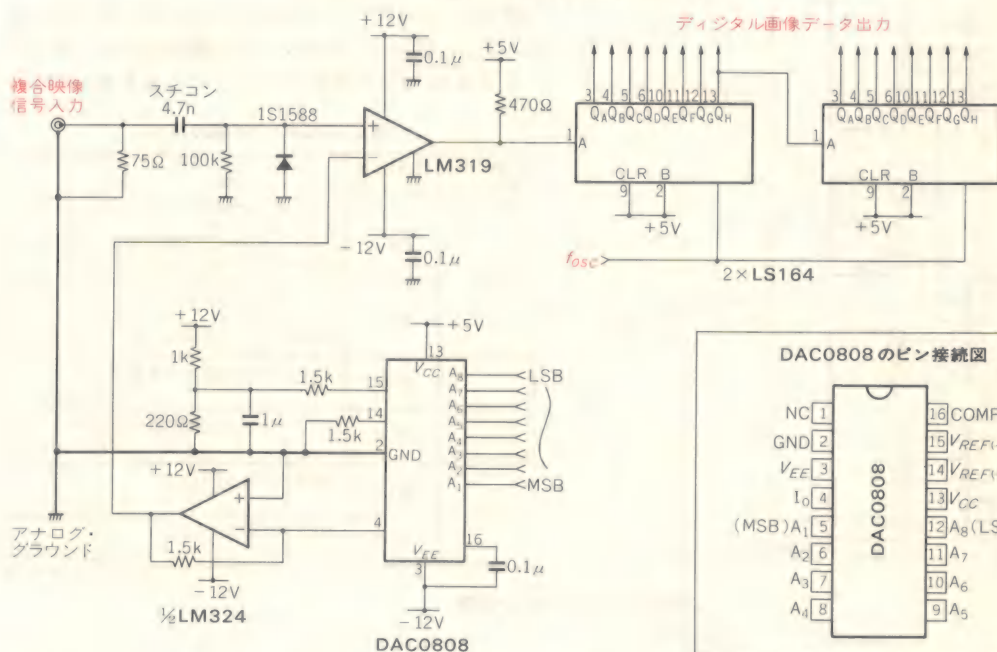
〈図11〉 サンプリグと量子化



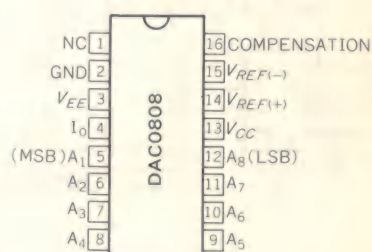
〈図12〉 サンプリグ・パルス発生回路



〈図13〉 映像信号のデジタル化



DAC0808のピン接続図



### ● 走査線カウンタ、水平位置カウンタ

走査線カウンタは、走査線の物理的な位置関係を示します。走査線を走査順で数えると、**上から、1、264、2、265...**となりますが、これは**1、2、3、4...**となります。

走査線カウンタは、10ビットのカウンタですが、上から9ビットはバイナリ・カウンタで、**LSB(最下位ビット)**はフィールド識別信号をそのまま使います。このことによって、1フィールドごとに、奇数・偶数の列が発生します。バイナリ・カウンタは、垂直同期

パルスでクリアされ、水平同期パルスでカウントされます。

水平位置カウンタは、1走査線中の位置を示すものです。ここでは、前述のシフトレジスタが16回シフトされるごとに、バイナリ・カウンタをインクリメントします。したがって、画像データの640ドットは、このカウンタの40に相当します。水平位置カウンタは、水平同期パルスでクリアされ、サンプリング・クロックを16分周したクロックによってカウントされます。

両カウンタによって、1フレームは先に示した図2



のように座標化されます。

両カウンタの出力は、デジタル・コンパレータに  
入力され、コンピュータから与えられた座標と比較さ  
れます。両方の値が一致したときに、座標一致パルス  
が出力されます。この回路図を図14に示します。

● ハンドシェイク回路の構成

サンプリングは、コンピュータからサンプリング・  
リクエスト信号を受けて開始され、画像情報のディジ  
タル化が終了すると、コンピュータに対してデータ・  
レディ信号を送ります。もし、1フレーム期間(1/30  
秒)以内にデジタル化に成功しなかった場合は、サン  
プリング・エラー信号を返します。以上が、ハンド

シェイク回路の動作です。

図15が実際のハンドシェイク回路です。コンピュ  
ータから与えられた座標と、走査線カウンタ、水平位置  
カウンタの示す座標とが一致すると、座標一致パルス  
が入力されます。この時点で、シフトレジスタの並列  
出力には画像データが現れていますので、データ・ラ  
ッチ・パルスでこれをラッチします。同じパルスでD  
型フリップフロップをトリガして、レディ信号をつく  
ります。

コンピュータが次のリクエスト信号を与えるまで、  
レディの状態は続きます。

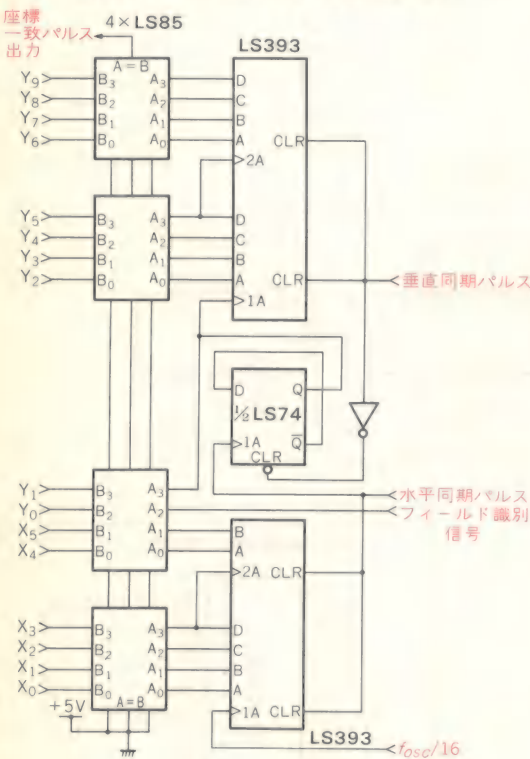
コンピュータが与えた座標が、実際にはフレームに  
存在しないような値であったときは、データ・ラッ  
チ・パルスは出力されません。このとき、エラー信号  
が出力されますので、ソフトウェアで、レディ信号と  
共にエラー信号を監視すると便利です。

REQ, RDY, ERRなどの信号線は、I/Oアドレス  
に割り付けておきます。

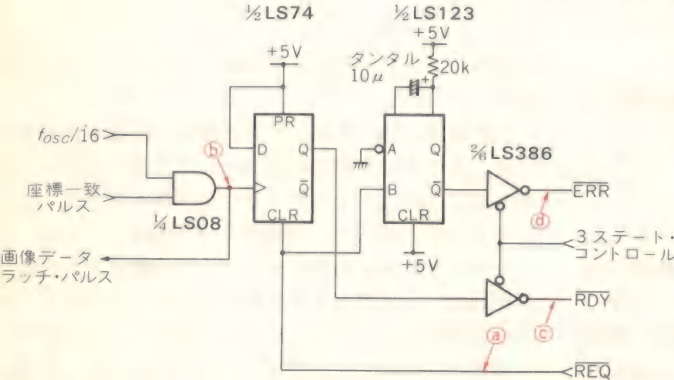
● I/Oポートの割り付け

表2に、I/Oアドレスの割り付けのようすを示し  
ます。また、図16に、I/Oポートの構成図を示します。  
ここではPC9801と接続することにしますが、8086を

〈図14〉 走査線カウンタ、水平位置カウンタ

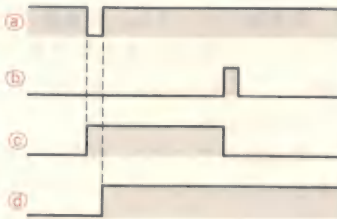


〈図15〉 ハンドシェイク回路



〈表2〉 I/Oポートの割り付け

I/O アドレス	CPUから 見た方向	機 能
DOH	IN	レディ信号 (ビット0)
		エラー信号 (ビット7)
	OUT	サンプリング要求
		D-Aコンバータ用ラッチ
D2H	IN	デジタル画像データ用ラッチ (下位バイト)
	OUT	サンプリング位置座標用ラッチ (下位バイト)
D3H	IN	D2HのINと同じ (上位バイト)
	OUT	D2HのOUTと同じ (上位バイト)

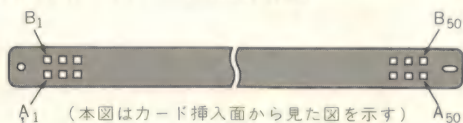




〈表3〉PC9801のバス・スロットの信号一覧

端子番号	信号名	方向	機能	端子番号	信号名	方向	機能
A <sub>1</sub>	GND			B <sub>1</sub>	GND		
A <sub>2</sub>	V1			B <sub>2</sub>	V1		
A <sub>3</sub>	V2			B <sub>3</sub>	V2		
A <sub>4</sub>	AB001	I/O	アドレス・バス	B <sub>4</sub>	DB001	I/O	データ・バス
A <sub>5</sub>	AB011	I/O	アドレス・バス	B <sub>5</sub>	DB011	I/O	データ・バス
A <sub>6</sub>	AB021	I/O	アドレス・バス	B <sub>6</sub>	DB021	I/O	データ・バス
A <sub>7</sub>	AB031	I/O	アドレス・バス	B <sub>7</sub>	DB031	I/O	データ・バス
A <sub>8</sub>	AB041	I/O	アドレス・バス	B <sub>8</sub>	DB041	I/O	データ・バス
A <sub>9</sub>	AB051	I/O	アドレス・バス	B <sub>9</sub>	DB051	I/O	データ・バス
A <sub>10</sub>	AB061	I/O	アドレス・バス	B <sub>10</sub>	DB061	I/O	データ・バス
A <sub>11</sub>	GND			B <sub>11</sub>	GND		
A <sub>12</sub>	AB071	I/O	アドレス・バス	B <sub>12</sub>	DB071	I/O	データ・バス
A <sub>13</sub>	AB081	I/O	アドレス・バス	B <sub>13</sub>	DB081	I/O	データ・バス
A <sub>14</sub>	AB091	I/O	アドレス・バス	B <sub>14</sub>	DB091	I/O	データ・バス
A <sub>15</sub>	AB101	I/O	アドレス・バス	B <sub>15</sub>	DB101	I/O	データ・バス
A <sub>16</sub>	AB111	I/O	アドレス・バス	B <sub>16</sub>	DB111	I/O	データ・バス
A <sub>17</sub>	AB121	I/O	アドレス・バス	B <sub>17</sub>	DB121	I/O	データ・バス
A <sub>18</sub>	AB131	I/O	アドレス・バス	B <sub>18</sub>	DB131	I/O	データ・バス
A <sub>19</sub>	AB141	I/O	アドレス・バス	B <sub>19</sub>	DB141	I/O	データ・バス
A <sub>20</sub>	AB151	I/O	アドレス・バス	B <sub>20</sub>	DB151	I/O	データ・バス
A <sub>21</sub>	GND			B <sub>21</sub>	GND		
A <sub>22</sub>	AB161	I/O	アドレス・バス	B <sub>22</sub>	+12V		
A <sub>23</sub>	AB171	I/O	アドレス・バス	B <sub>23</sub>	+12V		
A <sub>24</sub>	AB181	I/O	アドレス・バス	B <sub>24</sub>	IR31	I	INT <sub>0</sub>
A <sub>25</sub>	AB191	I/O	アドレス・バス	B <sub>25</sub>	IR51	I	INT <sub>1</sub>
A <sub>26</sub>	AB201	I/O	アドレス・バス	B <sub>26</sub>	IR61	I	INT <sub>2</sub>
A <sub>27</sub>	AB211	I/O	アドレス・バス	B <sub>27</sub>	IR91	I	INT <sub>3</sub> (5HD)
A <sub>28</sub>	AB221	I/O	アドレス・バス	B <sub>28</sub>	IR101/IR111	I	INT <sub>41</sub> /INT <sub>42</sub> (11)
A <sub>29</sub>	AB231	I/O	アドレス・バス	B <sub>29</sub>	IR121	I	INT <sub>5</sub>
A <sub>30</sub>	INT0	O		B <sub>30</sub>	IR131	I	INT <sub>6</sub>
A <sub>31</sub>	GND			B <sub>31</sub>	GND		
A <sub>32</sub>	IOCHK0	I	外部 NMI	B <sub>32</sub>	-12V		
A <sub>33</sub>	IOR0	I/O	コマンド	B <sub>33</sub>	-12V		
A <sub>34</sub>	IOW0	I/O	コマンド	B <sub>34</sub>	RESET0	O	RESET
A <sub>35</sub>	MRC0	I/O	コマンド	B <sub>35</sub>	DACK00	O	5' HD
A <sub>36</sub>	MWC0	I/O	コマンド	B <sub>36</sub>	DACK30/DACK20	O	AUX <sup>(1)</sup>
A <sub>37</sub>	S00	I/O	S0	B <sub>37</sub>	DRQ00	I	5' HD
A <sub>38</sub>	S10	I/O	S1	B <sub>38</sub>	DRQ30/DRQ20	I	AUX <sup>(1)</sup>
A <sub>39</sub>	S20	I/O	S2	B <sub>39</sub>	WORD0	I	
A <sub>40</sub>	LOCK0	I/O		B <sub>40</sub>	CPKILL0	I	
A <sub>41</sub>	GND			B <sub>41</sub>	GND		
A <sub>42</sub>	CPUENB10	O		B <sub>42</sub>	RQOT0	I	バスの解放要求
A <sub>43</sub>	RFSH0	O		B <sub>43</sub>	DMATC0	O	END OF PROCESS
A <sub>44</sub>	BHE0	I/O		B <sub>44</sub>	NMI0	O	
A <sub>45</sub>	IORDY1	I		B <sub>45</sub>	MWE0	O	
A <sub>46</sub>	SCLK1	O	7.9672/4.9152MHz <sup>(2)</sup>	B <sub>46</sub>	HLDA00	O	
A <sub>47</sub>	S18CLK1	O	307.2kHz	B <sub>47</sub>	HRQ00	I	
A <sub>48</sub>	POWER0	O	電源信号	B <sub>48</sub>	DMAHLD0	I	
A <sub>49</sub>	+5V			B <sub>49</sub>	+5V		
A <sub>50</sub>	+5V			B <sub>50</sub>	+5V		

(1) スロットφ1, φ2, φ3/スロットφ4 (2) 8 MHz/5 MHz



使った16ビット・マイコンであれば、デコーダを少し変更すれば接続できます。回路図を示す前に、PC9801の拡張スロットについて説明します。

PC9801の背面には、拡張スロットが用意されています。ここには表3に示すスロット・バスが出ています。普通はここに、増設RAMボードや固定ディスク・インターフェース・ボードを実装します。自作したインターフェースは、ユニバーサル・ボード(PC9801-04)に組むとよいでしょう。

ここで利用する信号線とその機能を、表4にまとめておきます。この中でBHE0について説明を加えます。これは、8086のBHEに接続されています。BHEとA<sub>0</sub>の組み合わせによって、アクセスするI/Oの選択を行います。

8086のI/O構成は、I/Oアドレスが0～65535で、それぞれが1バイトの大きさです。16ビットのワード・データは、任意の連続した2バイト単位で扱われます。表2のD2H, D3H番地には、それぞれ8ビットのポートが設けてあり、1バイトごとに2回のアクセスでデータを扱うことができます。

また、D2H, D3Hの計16ビットを1ワードとみなして、1回のアクセスでデータを転送することもできます。ワード単位でのアクセスは、偶数アドレスに対して行うのが時間的に有利です。

また、ワード単位でのアクセスのほうがバイト単位よりも有利で、機械語でのアクセスはそうすべきでしょう。バイト単位でのアクセスをせざるを得ないのは、BASICのINP関数やOUT命令を用いるときです。

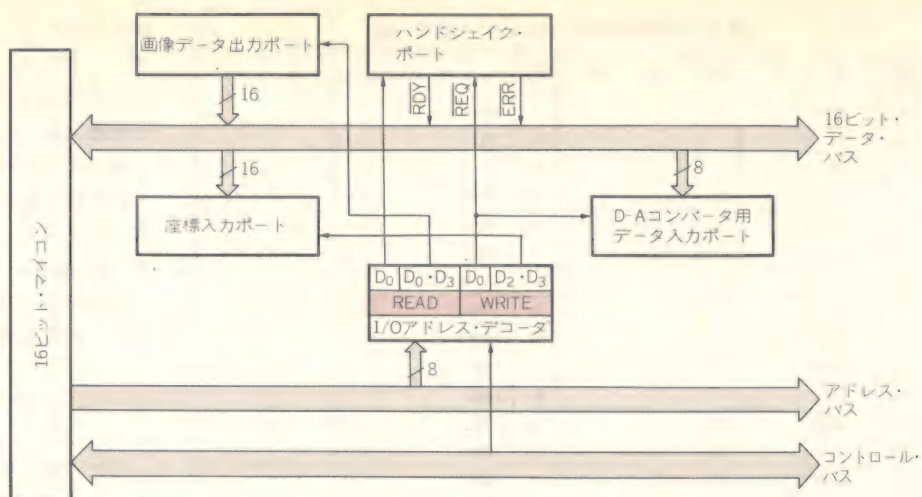
図17に、バイト転送をした場合と、ワード転送をした場合の、BHEとA<sub>0</sub>の組み合わせを示します。

I/Oポートの回路図を図18に示します。

図19に、PC9801用のカメラ・インターフェース・システムの全回路図を示します。図1のブロック図や、



〈図16〉  
I/Oポートの構成



〈表4〉使用するPC9801の信号線

信号名	機能
AB001 ↓ AB008	アドレス・バス
DB001 ↓ DB151	データ・バス
$\overline{IOR0}$	I/Oアクセスのリード・ストローブ
$\overline{IOW0}$	I/Oアクセスのライト・ストローブ
$\overline{BHE0}$	データ・バスの上位8ビットを有効にする
CPUENB10	CPUがバスを使用中にロー・アクティブ
+5V	1スロット最大 0.5A
+12V	1スロット最大 0.06A
-12V	1スロット最大 0.07A
GND	

前記の各部の説明と合わせて見れば理解しやすいと思います。写真1がインターフェース・ボードで、写真2がPC9801Fの後部スロットに挿入したところです。

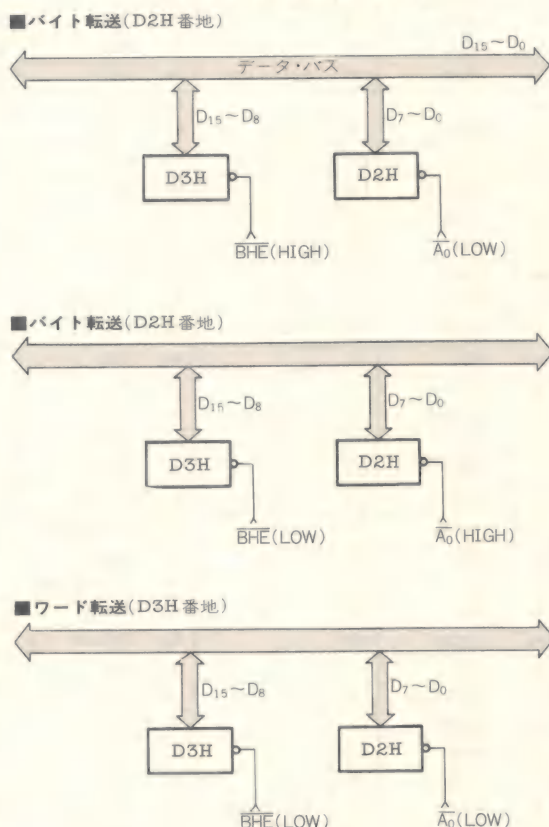
## ソフトウェアの設計

### ● ハンドラの機能

このハンドリング・プログラムの機能は、ひとことというと、フレームからVRAMへ画像データを転送することです。映像信号の1フレームは、ハードウェアによって図20のように座標化されています。この1フレームの中の縦400ライン・横40ワードがVRAMに納まる大きさです。PC9801では、縦400・横640ビットのグラフィック表示のとき、1画素が正方形になります。ここで紹介するハンドラには、次のような特徴があります。

- (1) フレームの中の、縦400ライン・横40ワードを、VRAMの640×400ドットの領域に転送する。

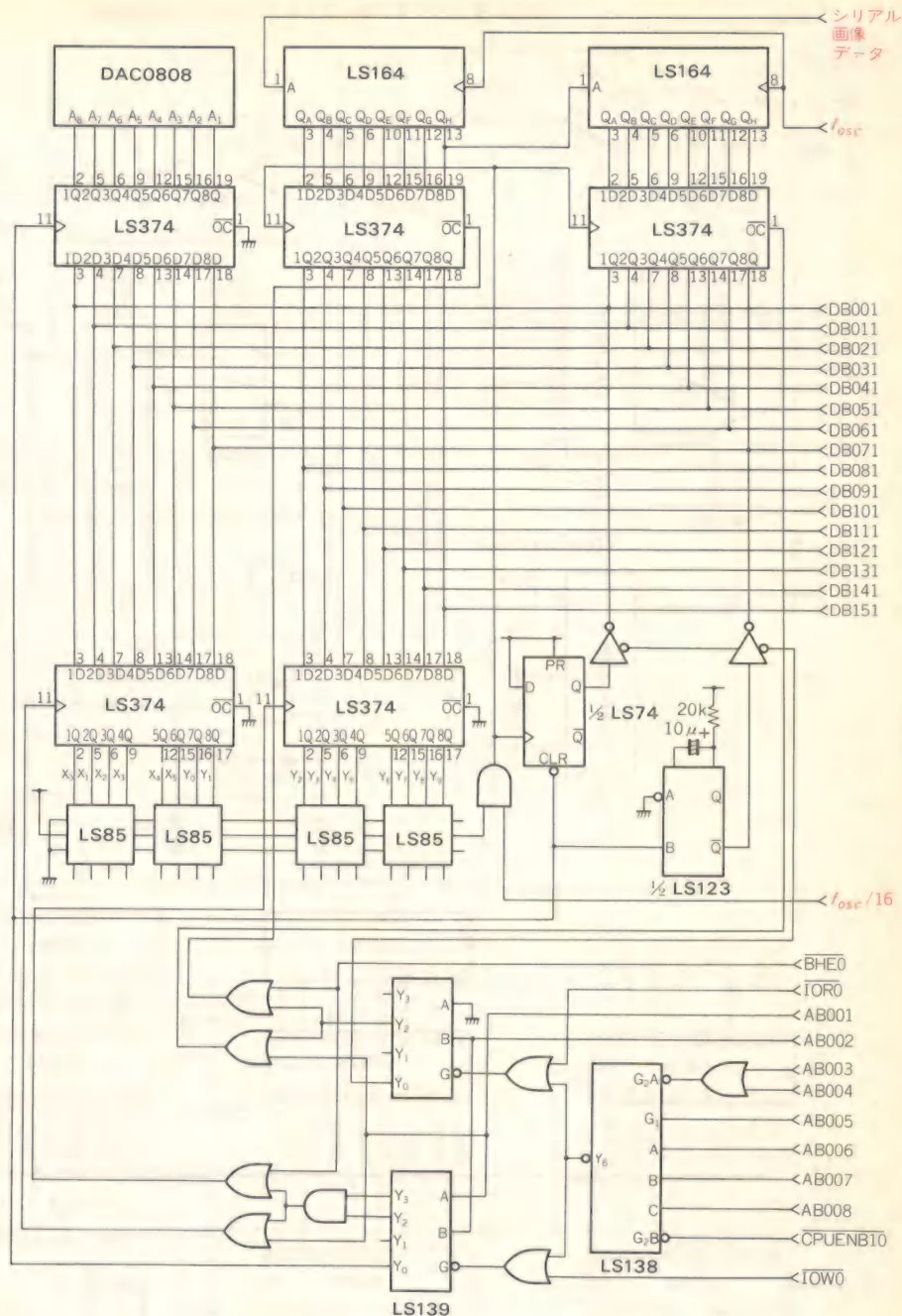
〈図17〉 バイト転送とワード転送



- (2) 上の範囲内で、フレームの部分領域を、VRAMの部分領域に転送することもできる。
- (3) 転送先のVRAMは、B・R・Gプレーンのいずれかを選択できる。
- (4) VRAMに転送するときのデータの置き方は、OR・XOR・AND・PSETのいずれかを選択できる。



〈図18〉  
I/Oポートの回路図



(5) 8階調入力のための、VRAM間転送ができる。B, R, Gプレーン相互間で転送される転送モードにはOR, XOR, AND, PSETがある。

#### ● ハンドシェイクについて

I/Oポートは、前述の表2のように割り付けてあります。データを入力する手順は次のようになります。

まず、図20のように、入力しようとする1ワードのデータの座標をセットします。次に、2値化レベルを設定するために、D-Aコンバータに適切な値を書き

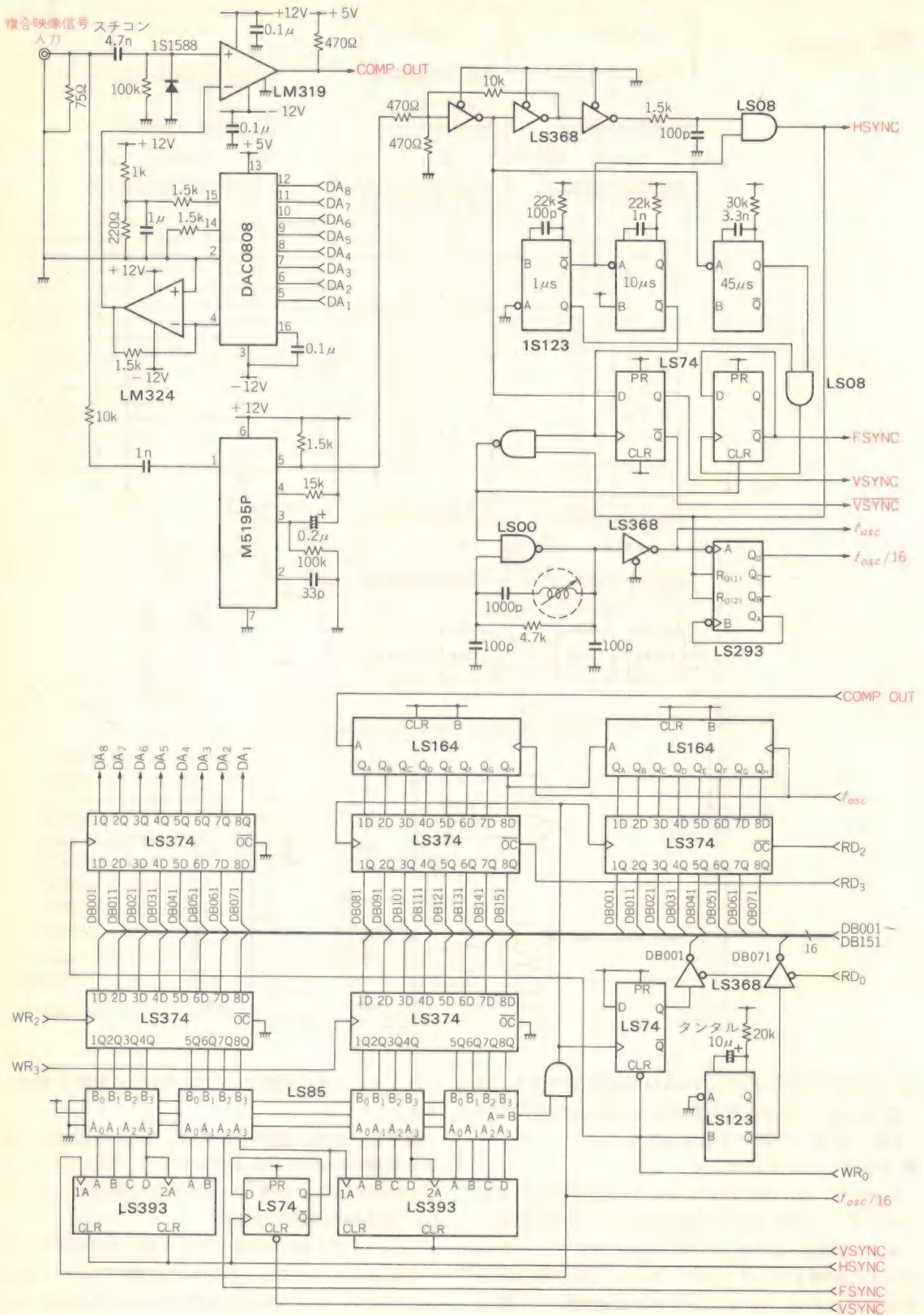
込みます。これは同時に、サンプリング要求を意味します。

指定した座標が適正であれば、1/30秒以内に、D 0 H番地のLSBが0となるはずですが、この~~レディ信号を確認したら、画像データをD 2 H番地からロードし、VRAMへ転送します。~~

もし、レディ信号を待っている間にMSBが0になった場合はエラーです。指定した座標がフレーム内がない場合や、カメラからの信号がNTSCに準拠してい

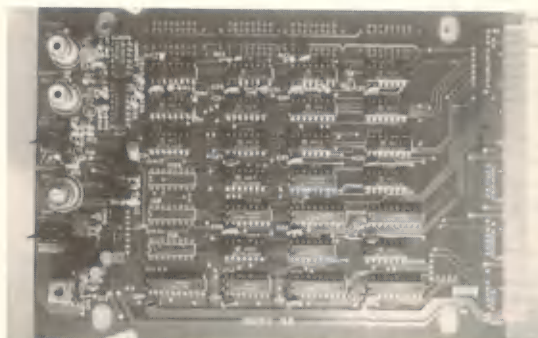
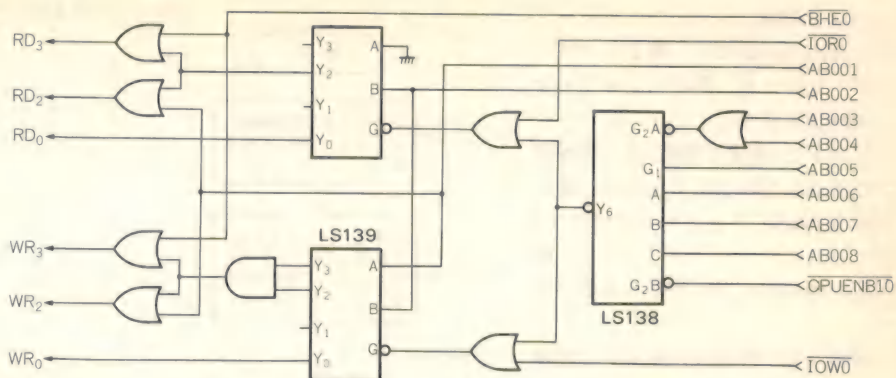


〈図19〉 テレビ・インターフェース・ボードの全回路図

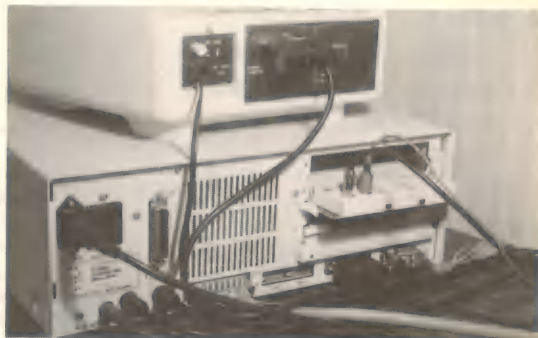




〈図19〉  
テレビ・インターフェース・ボードの全回路図  
(つづき)

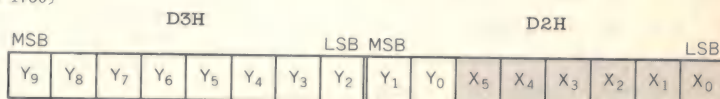


〈写真1〉完成したテレビ・インターフェース・ボード  
〔㈱テックメイト ☎03-792-1750〕



〈写真2〉PC9801のバスに挿入したところ

〈図20〉サンプリング位置座標の与え方



$$D3H \longleftarrow \text{INT}(Y/2^2)$$

$$D2H \longleftarrow X + ((Y \text{ AND } 3) * 2^6)$$

Y, Xは、図2の水平位置カウンタ値と  
走査線カウンタ値を示す。

ない場合や、ハードウェアが不具合な場合などが原因として考えられます。

### ● サンプリングの方法

上に述べたことを繰り返せばよいわけですが、指定する座標の与え方によって、VRAM 1画面の入力に要する時間が決まります。サンプリング要求をして、レディ信号が出るまで待つ時間が短いほど、入力時間は短くなります。

CPUのデータ取り込みの時間間隔が最小になるように、座標を次々と与えるためには、あらかじめ与える総座標を算出するプログラムによって座標テーブルを作り、順次与えてはデータ入力をする。という方法があります。しかし、ここではスピードは多少遅くなりますが、理解しやすい次のような方法を紹介し

それは、1水平期間に1ワード入力する方法です。1水平期間の63.5μsがサンプリングの周期となり、PC9801の場合、5MHzで動作させても十分間に合います。640×400のVRAMに転送するのに要する時間は、40フレーム期間の約1.33秒です。与える座標は、規則的に発生すればよく、プログラムはやさしくなります。

### ● ハンドリング・プログラム

リスト1(章末に掲載)に、その一例を示します。PC9801のCPUである8086の機械語です。CP/M-86に付属のASM86でアセンブルできるソースになっています。図21に、入力に関するゼネラル・フローチャートを示します。

この機械語は、N<sub>88</sub>(86)BASICから呼ばれ、パラメ



ータはBASICとインターフェースするための引数です。表5にパラメータ・リストを、図22にメモリ・マップを示します。

リスト2(章末に掲載)は、機械語プログラムを呼び出すための、最も簡単なBASICプログラムの例です。表5とこのプログラムによって、機械語の呼び出し方がわかるといいます。

写真4は、写真3の原画を2階調で入力したものです。

● 階調付き入力プログラム

この装置は、基本的には映像信号を2値化して、デジタル映像信号とするものですが、2値化レベルを変えながら数度にわたって入力することで、階調性を表現することができます。

例えば、第1回目は、青プレーンに入力します。次に2値化レベルをやや白レベルのほうに移し、赤プレーンに入力したあと、赤プレーンから青プレーンに向かってXORで転送します。第3回目は青プレーンにORで入力します。

ここまでで、一番暗いところが青で、次が赤で、一番明るいところが紫になっています。これをモノクロ・ディスプレイで見ると、色番号の順が明るさの順に見えます。

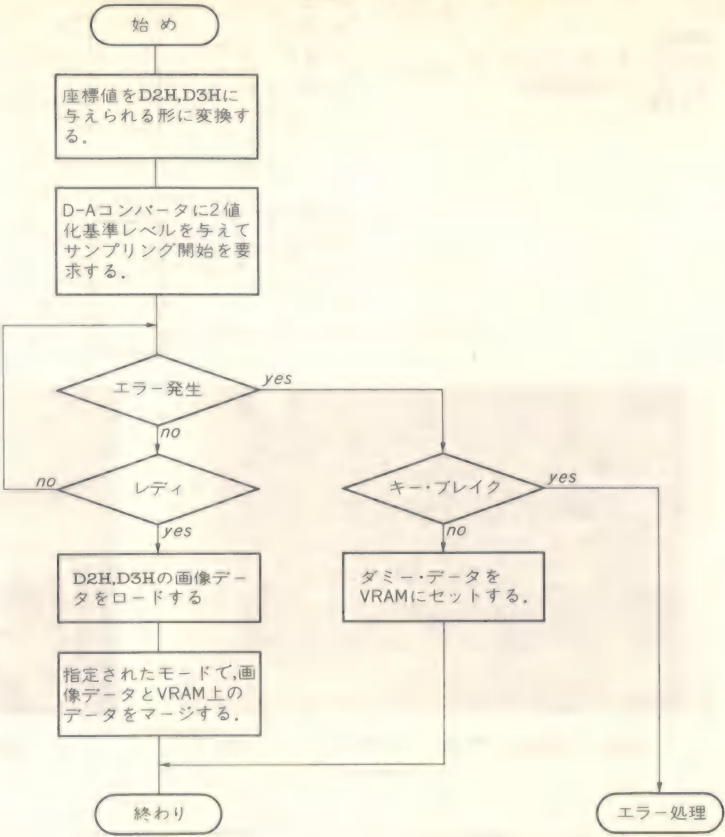
このように7回入力すると、黒を含めた8階調が表現できます。

リスト3(章末に掲載)が、階調入力のできるプログラムの例で、2階調のほか、4階調、8階調の入力ができます。

プロンプト行の、“LEVEL”というのが2値化レベルで、右上のローマ数字が階調を表しています。入力は[Y]キーまたは[RET]キーで開始し、2階調の場合は[ESC]キーで終了し、4、8階調の場合は自動的に終了します。左右カーソル・キーで2値化レベルの設定ができ、シフト・キーとの併押で10飛びに変化します。2値入力の最中などでもレベル変更は可能です。

写真5は、写真3の原画を8階調

〈図21〉 データ入力のフローチャート



〈表5〉 変数の意味

	パラメータ名	ビット長	機 能
入力用 パラ メー タ	inpseg	16	入力する VRAM のセグメント
	bgnadr	16	VRAM の入力エリアの左上のオフセット・アドレス
	vbg	16	入力を開始する走査線番号
	yleng	16	VRAM の入力エリアの縦の長さ
	xleng	8	VRAM の入力エリアの横の長さ(ワードで指定)
	inpmod	8	VRAM データをセットするときのモード 0:or, 1:xor, 2:and, 3:pset
	hbgn	8	入力を開始する水平位置
	dadat	8	D-A コンバータに与える 2 値化レベル
	subcode	8	0:インプット, 1:VRAM 間転送
	retcd	8	BASIC に返す引数 0:正常に入力が完了した, 1:入力中にエラーが発生した, 2:入力中にエラーが発生し、かつそのときにキー 割り込みがあった。
転送用 パラ メー タ	yscrn	8	0:高解像度 CRT, 1:低解像度 CRT
	srcseg	16	ソース・データのあるセグメント
	dstseg	16	デスティネーションのセグメント
	sttadr	16	転送エリアの左上のオフセット・アドレス
	ywidth	16	転送エリアの縦の長さ
	xwidth	8	転送エリアの横の長さ(ワードで指定)
	xmode	8	転送データをセットするときのモード 0:or, 1:xor, 2:and, 3:pset
	subcode	8	0:インプット, 1:VRAM 間転送





〈写真3〉入力した原画



〈写真4〉2値入力画面



〈写真5〉8階調入力画面

入力したものです。

### ● おわりに

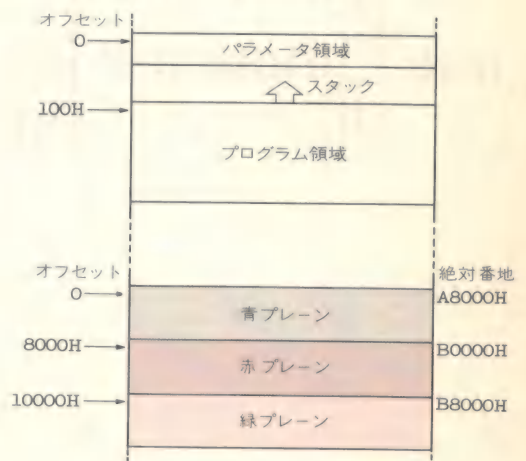
テレビ・カメラとPC9801を容易にインターフェースする方法を紹介しました。ここでは、PC9801をホスト・コンピュータとして使用しましたが、8086をCPUとするほかの機種との接続も容易です。

カラー・カメラとのインターフェースや、入力時間を短縮する方法や、色々な応用については、ここに述べてきたインターフェースをベースにして可能ですので、機会がありましたらまた紹介したいと思います。

### ● 参考文献

- (1) ソニー, XC-37カタログ.
- (2) 日立, CCTV装置カタログ.
- (3) 日立, リモートアイカタログ.
- (4) テックメイト, カメラ・インターフェース・システム仕様書.
- (5)\*NEC, PC9801ユーザーズ・マニュアル.
- (6)\*三菱, データブック テレビ/ビデオ編, 1983年.
- (7)\*NS, リニア・データブック, 1980年.
- (8) 水野博之; オプトエレクトロニクス, 日刊工業新聞社.
- (9) 木内雄二; 電子画像のはなし, 日刊工業新聞社.
- (10) 木内雄二; イメージセンサ, 日刊工業新聞社.
- (11) 井出裕巳; 8086の使い方, オーム社.
- (12) 木下健治; 画像処理システムの基礎と設計・製作, CQ出版社.
- (13) 尾崎, 谷口, 小川; 画像処理, 共立出版.
- (14) 高橋恒雄; 静止画像入力装置の製作, トランジスタ技術, 1983年2月号.
- (15) ビデオカメラ最新ガイド, 別冊電波科学, 日本放送出版協会.
- (16) 佐藤, 林; 画像伝送とその機器, コロナ社.
- (17) 長谷川伸; 画像工学, コロナ社.

〈図22〉メモリ・マップ





# リスト1) 機械語によるハンドラ部

```

;*****
; * PROGRAM: CAMERA INTERFACE SYSTEM HANDLER *
; * VERSION 0.00 BY UNO *
;*****
;*** definition of constants ***
datofst equ 0000h ;data offset address
cdofst equ 0000h ;code offset address
stktop equ cdoftst-2 ;stack top
keyptop equ 43h ;serial key port
;
aseq
org
;*** parameters for input ***
inseq dw 0000h ;segment of input vram
panadr dw 0000h ;left upper address of input area
vbn dw 0000h ;scan line number
;
vlen dw 0000h ;height of input area (1*400)
xlen dw 00h ;width of input area (1*40)
;
inmod db 00h ;data set mode: 0...or 1...xor
;
nbn db 00h ;horizontal position number
;
dadat db 00h ;
subcode db 00h ;
retcd db 00h ;
;
vscrn db 00h ;
;
;*** parameter for transfer ***
srcseq dw 0000h ;source vram segment
dstseq dw 0000h ;destination vram segment
stadr dw 0000h ;left upper address of transfer area
width dw 0000h ;height of transfer area
xwidth dw 00h ;width of transfer area
xmode db 00h ;data set mode: same as inmod
;
;*** refude of registers ***
ssave dw 0000h
ssave dw 0000h
;
;
cseq
org
;begin of code
;
;*****
; save registers and jump
;*****
c11
mov ax,cs
mov ds,ax
mov ssave,ss
mov ssave,sp
mov ss,ax
mov sp,stktop
;
push ds
push es
push di
push si
push bp
push dx
push cx
push ax

```

①

```

;
mov bh,0
mov bl,subcode
rol bx,1
add bx,offset jmotbl
mov ax,[bx]
jmp ax ;jump by subcode value
;
;*****
; resave registers and back to BASIC
;*****
return: pop ax
pop cx
pop bx
pop bp
pop si
pop di
pop es
pop ds
;
mov ss,sssave
mov sp,sssave
xor ax,ax
sti
iret
;
;*****
; make dma free
; return to basic
;*****
; jump table
;*****
jmotbl dw offset input ;input picture
; dw offset xfer ;transfer vram data
;
;*****
; execute patch
; in order to change data set mode
;*****
patch: mov al,inmod
; mov al,00h ;or?
; cmp al,01h ;xor?
; jz xormod ;and?
; cmp al,02h ;and?
; jz andmod ;tost?
; cmp al,03h ;tost?
; jz psetmod
;
; ormod: mov al,09h ;patch data
; jmps patch2
; xormod: mov al,31h ;patch data
; jmps patch2
; andmod: mov al,21h ;patch data
; jmps patch2
; psetmod: mov al,89h ;patch data
; patch2: mov cx,0,al ;patch done
;
;
; cmd vscrn,0
; jz v400
;
; v200: mov bx,9090h ;select 200 line input
; mov al,50h
; mov ah,90h
; jmps patch3
;
; v400: mov bx,0f60h ;select 400 line input
; mov al,0a0h
; mov ah,0edh

```

②



```

: patch3: mov    pch1,al      ;patch done
:          mov    pch2,ah
:          mov    pch3,bx
:          ret
:
: input one field
: ( 200 lines with 16 bits width )
: vinout: mov    dx,vieng
:
: vinloop: or     dx,dx
:          jz     vinend
:
: xform:  mov     ah,bl
:          mov     al,bh
:          shr     al,1
:          rcr     ah,1
:          rcr     al,1
:
:          mov     al,bl
:          shl     al,1
:          shl     al,1
:          shl     al,1
:          shl     al,1
:          shl     al,1
:          add     al,cl
:
:          out     0d2h,ax
:
: request: mov     al,dadat
:          out     0d0h,al
:          wait:   in     al,0d0h
:          test    al,80h
:          jz     error
:          jmp    ierror
:          test    al,01h
:          jnz     wait
:
:          in     ax,0d2h
:
:          setb    %
:          db     26h
:          pch0    db     09h
:          db     04h
:
:          db     0b8h
:          db     0a0h
:          db     00h
:          add     si,ax
:
:          inc     bx
:          inc     dx
:          dec     dx
:          jmp     vinloop
:
: vinend: ret
:
: error:   mov     retcd,1
:          in     al,keyboot
:          and     al,02h
:          erct    jz
:          mov     mov    pch2,ax
:          pop     return

```

③

```

: erct:    mov     ax,0xffffh ;load dummy data
:          jmps    setd
:
:          : input main
:          :
:          input:  mov     retcd,0
:                  call    patch
:                  mov     ch,xlend
:                  mov     ax,bonadr
:                  mov     si,ax
:                  mov     ax,inpseq
:                  mov     es,ax
:                  mov     cl,hagn
:
:          : inloop: or     ch,ch
:                  jz     inpend
:
:                  mov     bx,vagn
:                  mov     dx,si
:
:                  : call    vinout
:                  mov     bx,vagn
:                  inc     dx
:                  mov     si,db
:                  mov     ax,0050h
:                  add     si,ax
:
:                  : pch2
:                  db     0e8h
:                  pch3    dw     0f6bh
:
:                  : call    vinout
:                  mov     ax,db
:                  inc     ax
:                  inc     ax
:                  mov     si,ax
:                  inc     cl
:
:                  : dec     ch
:                  jmps    inloop
:
:          : inpend: jmp     return
:
:          : patch in order to change dot set mode
:          :
:          xoch:    mov     al,xmode
:                  or     al,al
:                  jz     orset
:                  cmp     al,01h
:                  jz     xorset
:                  cmp     al,02h
:                  jz     andset
:
:                  :
:                  dbset:   mov     al,89h
:                  jmp     pcheke
:                  andset:  mov     al,21h
:                  jmp     pcheke
:                  xorset:  mov     al,31h
:                  jmp     pcheke
:                  orset:   mov     al,09h
:                  pcheke:  mov     pchdatz,al
:                  ret

```

④



（リスト 2）機械語を読み出すBASICプログラム

```

:=====
: xter one vertical line
: width to bits width
:=====
:vert: push ds
:      mov cx,width
:      mov ax,rcxseq
:      mov ds,ax
:
: xvlod: mov ax,ds:[s1]
:      db 26h
:      ovd2 db 07h
:      db 04h
:
:      mov ax,0050h
:      add ax,s1
:      mov si,ax
:
:      lod word ptr [si]
:      mov dx,ax
:      ret
:
:=====
: xter main
:=====
: xter: call xoch
:
:      mov ax,dstseq
:      mov es,ax
:
:      xor cx,cx
:      mov cx,width
:      mov ax,xtadr
:      mov si,ax
:
: xtlod: push si
:      push cx
:      call xvert
:      pop cx
:      pop si
:      inc si
:      lod word ptr [si]
:
:      jmp return
:
:      end
:
: c:

```

```

1000 CLEAR %HIF80:WIDTH 80.25:CONSOLE 0.25.0.1:CLS 2
1010 DEF SEG=%HIF80 :BLOAD "HANDLER"
1020 DEF USR0=%H100
1030
1040 GOSUB *DEF.PARA
1050 IF YSCRN=0 THEN SCREEN 3.0.0.1 ELSE SCREEN 0.0.0.1
1060 GOSUB *POKE.PARA
1070 EXEC=USR0(0) :RETC0=PEEK(13)
1080 IF RETC0=0 THEN PRINT "Input End"
1090 IF RETC0=1 THEN PRINT "Input Error"
1100 IF RETC0=2 THEN PRINT "Key Break"
1110 END
1120
1130 *DEF.PARA
1140 YSCRN=0
1150 SEGH=%HAB :SEGL=0
1160 BGNADR=80*0+0
1170 YLENG=400
1180 XLENG=40
1190 INPMOD=3
1200 VBGN=80
1210 HBGN=4
1220 DADR=70
1230 SUBCODE=0
1240 RETURN
1250
1260 *POKE.PARA
1270 POKE 0,SEGL
1280 POKE 2,BGNADR MOD 256
1290 POKE 4,VBGN MOD 256
1300 IF YSCRN=0 THEN
    POKE 6,(YLENG #2) MOD 256 :POKE 7,(YLENG # 2) # 256
    ELSE POKE 6,YLENG MOD 256 :POKE 7,YLENG # 256
1310 POKE 8,XLENG
1320 POKE 9,INPMOD
1330 POKE 10,HBGN
1340 POKE 11,DADR
1350 POKE 12,SUBCODE
1360 POKE 14,YSR0
1370 RETURN

```



### (リスト3) 階調入力プログラム

```

1000 *****
1010 * CAMERA INTERFACE SYSTEM (PC-9801/F/E) *****
1020 * VER 0.00 BY UNO *
1030 *****
1040 GOTO 1230
1050 CONSOLE 0.25,0.1:WIDTH 80.25:COLOR 7
1060 DEF SEG=MASEG:BLORD "HANDLER"
1070 GOSUB 1410:GOSUB 1130:SCREEN SCRNMOD,0,0,1
1080 YL=YSCRN-1:IF YSCRN=400 THEN YT=22 ELSE YT=11
1090 DEFAULT.DA=70:DA=DEFAULT.DA:DA.WINDOW=60:GRADSAVE=8
1100 GOTO 2420:go to main
1120
1130 ***** Hard ware check *****
1140 POSH=10*2^2:POSL=((10 AND 3)*2^6)+10
1150 OUT &HD2,POSL:OUT &HD3,POSH:OUT &HD0,255
1160 FOR DLY=0 TO 1000
1170 IF (INP(&HD0) AND &HB0)=0 THEN DLY=1000
1180 NEXT DLY
1190 IF (INP(&HD0) AND &HB0)=0 THEN RETURN
1200 GOSUB 1540:SCREEN SCRNMOD,0,0,0:CLS
1210 LOCATE 15.15:PRINT "Camera interface system is not on-line."
:END
1230 ***** Kanji ROM & memory size check *****
1240 DEF SEG=&HA3FE:MSW3=PEEK(&HA)
1250 MSIZE=MSW3 AND 7
1260 ON MSIZE+1 GOTO 1270,1280,1290,1300,1310
1270 CLEAR:&H1F80:MASEG=&H1F80:GOTO 1320
1280 CLEAR:&H3F80:MASEG=&H3F80:GOTO 1320
1290 CLEAR:&H5F80:MASEG=&H5F80:GOTO 1320
1300 CLEAR:&H7F80:MASEG=&H7F80:GOTO 1320
1310 CLEAR:&H9F80
1320 DEF SEG=MASEG:DEF USK0=&H100
1330 DEF SEG=&HAB00:DIM WD%(16):SUM=0
1340 GOSUB 1410:IF YSCRN=200 THEN 1050
1350 PUT(0,0).KANJ1(&H2130).PSET,7,0
1360 FOR WD=0 TO 15:SUM=SUM+PEEK(WD*80):NEXT WD:LINE(0,0)-(15,15)
:OF
1370 IF SUM>63 THEN GOTO 1050
1380 CHAIN MERGE "KANJ1.DAT",15,ALL,DELETE 1500
1390 GOTO 1050
1400
1410 ***** CRT type check *****
1420 ON ERROR GOTO 1470
1430 SCREEN 3,0,0,0:YSCRN=400:SCRNMOD=3:GOTO 1450
1440 SCREEN 0,0,0,0:YSCRN=200:SCRNMOD=0
1450 ON ERROR GOTO 0
1460 RETURN
1470 RESUME 1440
1480

```

```

1490 ***** Decide return code from M.C. *****
1500 IF RETCD=0 THEN RETURN
1510 IF RETCD=1 THEN RETURN
1520 IF RETCD=2 THEN GOSUB 1550:PRINT "User Key Break":END
1530 STOP
1540 FOR BEE=0 TO 6:BEEP(BEE MOD 2):FOR DLY=0 TO 100:NEXT:RETURN
1550 GOSUB 1540:FOR I=0 TO 7:COLOR=(1,1):NEXT I:SCREEN SCRNMOD,0,0,0:CLS
1560 LOCATE 30.23:PRINT "Hit ("CHR$(165):"S) (RUN)":LOCATE 30.11:RETURN
1570
1580 ***** Give data to DA-converter *****
1590 IF C$="INC" THEN DA=DA+CC
1600 IF C$="01R" THEN DA=CC:GOTO 1660
1610 IF DA>255 THEN DA=255 ELSE IF DA<0 THEN DA=0
1620 IF GRAD=2 THEN 1660
1630 IF DA<-1 AND DA<DA.WINDOW#2+1 THEN DA=DA.WINDOW#2
1640 IF DA<256 AND DA>256-DA.WINDOW#2 THEN DA=256-DA.WINDOW#2-1
1650
1660 GOSUB 1690:LOCATE 62,0:PRINT "LEVEL":LOCATE 67,0:PRINT USING "###":DA:
1670 OUT &HD0,DA:POKE 11,DA
1680 RETURN
1690
1700 LINE(0,0)-(192-1,0):2:LINE(192,0)-(192+256,0):0:LINE(192+256+1,0)-
(639,0):2
1710 LINE(192,0)-(192+DA-1,0):7,.&HCCCC
1720 LINE(192+DA,0)-(192+256,0):7
1730 RETURN
1740
1750 ***** Transfer VRAM data to other one *****
1760 IF I=2 THEN SRCSEG=&H80:DSTSEG=&HAB:XMODE=1:GOSUB 1800:RETURN
1770 IF I=4 THEN SRCSEG=&H8B:DSTSEG=&HB0:XMODE=1:GOSUB 1800:DSTSEG=&HAB
:GOSUB 1800:RETURN
1780 RETURN
1790
1800 ***** Transfer sub *****
1810 STIADR=0+YI*80:XWIDTH=40:YWIDTH=YSCRN-YT:SUBCODE=1
1820 POKE &H1F,0:POKE &H10,SRCEG:POKE &H11,0:POKE &H12,DSTSEG
1830 POKE &H13,STIADR MOD 256:POKE &H14,STIADR # 256
1840 POKE &H17,XWIDTH:POKE &H15,YWIDTH MOD 256:POKE &H16,YWIDTH # 256
1850 POKE &H18,XMODE:0=OR:1=XOR:2=AND:3=SET
1860 POKE &HC,SUBCODE:MC=USR0(0)
1870 RETURN
1880
1890 ***** Select input mode *****
1900 IF GRAD=8 THEN GRADSAVE=GRAD:LOCATE 75,0:PRINT " 11":GRAD=2:RETURN
1910 IF GRAD=2 THEN GRADSAVE=GRAD:LOCATE 75,0:PRINT " 1V":GRAD=4:RETURN
1920 IF GRAD=4 THEN GRADSAVE=GRAD:LOCATE 75,0:PRINT "V11":GRAD=8:RETURN
1930 STOP
1940
1950 ***** Real time key input *****
1960 BRK=0
1970 AS=INKEY$
1980 IF AS="" THEN 1980:PRINT "CHR$(%H10): ELSE RETURN
1990 IF BRK=1 THEN PRINT "CHR$(%H10):RETURN
2000 IF AS="" THEN 2000:PRINT "CHR$(%H10): ELSE RETURN
2010 IF BRK=1 THEN RETURN
2020 GOTO 1970

```



```

2030 FOR DLY=0 TO 100
2040 AS=INKEY$: IF AS<>" THEN BRK=1 : DLY=100
2050 NEXT DLY
2060 RETURN
2070
2080 ----- Select write vram -----
2090 IF 1=1 THEN SEG=H8:GOTO 2160
2100 IF 1=2 THEN SEG=H8:GOTO 2160
2110 IF 1=3 THEN SEG=H8:GOTO 2160
2120 IF 1=4 THEN SEG=H8:GOTO 2160
2130 IF 1=5 THEN SEG=H8:GOTO 2160
2140 IF 1=6 THEN SEG=H8:GOTO 2160
2150 IF 1=7 THEN SEG=H8:GOTO 2160 ELSE STOP
2160 POKE 7,SEG : RETURN
2170
2180 ----- Input picture sub -----
2190 IF 1=6 THEN INPMOD=1 ELSE INPMOD=0 : 0=OR : 1=XOR
2200 IF GRAD=2 THEN INPMOD=3 : 0=OR,1=XOR,2=AND,3=PSET
2210 AS="INC":GOSUB 1580
2220 VBN=80 : HBN=4 : SUBCODE=0
2230 BGNADR=0+Y1*80 : YLENG=YSCRN-Y1 : XLENG=40
2240 DIV=YSCRN/200
2250 POKE 9,INPMOD : POKE 14,0 : POKE 0,0 : POKE 1,SEG
2260 POKE 2,BGNADR MOD 256 : POKE 3,BGNADR # 256
2270 POKE 6,(YLENG#DIV) MOD 256 : POKE 7,(YLENG#DIV) # 256 : POKE 8,XLENG
2280 POKE 10,HBN : POKE 4,VBN MOD 256 : POKE 5,VBN # 256
2290 IF YSCRN=400 THEN POKE 14,0 ELSE POKE 14,1
2300 POKE 15,SUBCODE : USR(0) : RETCD=FEEL(13) : GOSUB 1490
2310 IF RETCD=1 THEN GOSUB 2330
2320 RETURN
2330 SCREEN SCRNMOD,0,0,LOCATE 30,11:PRINT "Input Error":
2340 FOR DLY=0 TO 3000:NEXT DLY
2350 LOCATE 0,11:PRINT SPACE$(80)::SCREEN SCRNMOD,0,0,1:RETURN
2360
2370 ----- Clear line -----
2380 LOCATE 25,0:PRINT SPACE$(55):LOCATE 25,0:RETURN
2390 LOCATE 0,0:PRINT SPACE$(80) : LOCATE 0,0:RETURN
2400 LOCATE 25,0:PRINT SPACE$(46):LOCATE 25,0:RETURN
2410
2420 ***** INPUT PICTURE MAIN *****
2430 GOSUB 2390:FOR C=0 TO 7:COLOR=(C,C):NEXT C
2440 LOCATE 0,0:PRINT "*** INPUT NEW PICTURE ***":
2450 BRAD=BRADSAVE:TIMES=0:GOSUB 1890
2460
2470 C$="":GOSUB 1580
2480 B$=INKEY$:B$=INKEY$:B$=INKEY$:B$=INKEY$:B$=INKEY$
2490 LOCATE 25,0:PRINT "COMMAND = Y V <- -> . Esc ? ":GOSUB 1950
2500 IF AS="Y" OR AS="V" OR AS="." OR AS="<" OR AS=">" THEN ROLL 1:ROLL YL:IF GRAD=2
THEN GOTO 2820 ELSE GOSUB 2580 :GOTO 2470
2510 IF ASC(AS)=28 THEN C$="INC" : IF (INP(&H8) AND &H40)=64 THEN CC=1
:GOSUB 1580 ELSE CC=10:GOSUB 1580
2520 IF ASC(AS)=29 THEN C$="INC" : IF (INP(&H8) AND &H40)=64 THEN CC=-1
:GOSUB 1580 ELSE CC=-10:GOSUB 1580
2530 IF ASC(AS)=27 THEN GOTO 2970
2540 IF ASC(AS)=1 THEN GOSUB 2820 :GOTO 2420
2550 IF AS="V" OR AS="." OR AS="<" OR AS=">" THEN GOSUB 1890
2560 GOTO 2470
2570
2580 ----- Input 100 -----
2590 DASAVE=DA : DA,MIN=DA-DA,WINDOW # 2
2600 IF GRAD=2 THEN DA,MIN=DA:COLOR=(1,7):LAST,PIC=2
2610 IF GRAD=4 THEN DA,STEP=DA,WINDOW # 2:COLOR=(1,1):LAST,PIC=4
2620 IF GRAD=8 THEN DA,STEP=DA,WINDOW # 6:COLOR=(1,1):LAST,PIC=8
2630
2640 FOR J=1 TO BRAD-1
2650 C$="DIR":LC=DA,MIN+DA,STEP*(J-1):GOSUB 1580
2660 IF GRAD=2 THEN I=0 :GOTO 2690
2670 IF GRAD=4 THEN I=J :GOTO 2690
2680 IF GRAD=8 THEN I=J ELSE STOP
2690 GOSUB 2080
2700 IF I=6 THEN INPMOD=1 ELSE INPMOD=0
2710 GOSUB 2180
2720 IF I=3 THEN SEG=H8:INPMOD=0:GOSUB 2180
2730 IF I=5 THEN SEG=H8:INPMOD=0:GOSUB 2180
2740 IF I=6 THEN SEG=H8:INPMOD=1:GOSUB 2180
2750 IF I=7 THEN SEG=H8:INPMOD=0:GOSUB 2180:SEG=H8:GOSUB 2180
2760 GOSUB 1750
2770 NEXT J
2780
2790 DA=DASAVE
2800 RETURN
2810
2820 ----- Input binary picture -----
2830 GOSUB 2400:PRINT "COMMAND = <- -> . Esc ?":
2840 TIMES=1 :GOSUB 2580
2850 AS=INKEY$:FOR I=1 TO 32:B$=INKEY$:NEXT I
2860 IF AS=CHR$(28) THEN C$="INC" : IF (INP(&H8) AND &H40)=64 THEN
CC=1:GOSUB 1580 ELSE CC=10:GOSUB 1580
2870 IF AS=CHR$(29) THEN C$="INC" : IF (INP(&H8) AND &H40)=64 THEN
CC=-1:GOSUB 1580 ELSE CC=-10:GOSUB 1580
2880 IF AS=CHR$(27) THEN GOSUB 2910 :GOTO 2470
2890 GOTO 2840
2900
2910 IF TIMES=0 THEN 2940
2920 COLOR=(3,0)
2930 SRCSEG=&HAB:DSTSEG=&HB0:XMODE=1:GOSUB 1800:DSTSEG=&HB8:GOSUB 1800
2940 FOR C=0 TO 7:COLOR=(C,C):NEXT C
2950 RETURN
2960
2970 *** END ***
2980 GOSUB 2390:LINE(0,0)-(639,YT-1),0,BF:WIDTH 80,25:COLOR 7
2990 GOSUB 2390:PRINT "*** END ***"
3000 GOSUB 2380:PRINT"OK ? (Y/K/Esc) ":
3010 GOSUB 1950
3020 IF AS="Y" OR AS="." OR AS="<" OR AS=">" THEN CLOSE:SCREEN ,2,0,1:WIDTH 80,25:CONSOLE
0,25,1:END
3030 IF AS="K" OR AS="." THEN CLOSE:WIDTH 80,25:CONSOLE 0,25,0:END
3040 GOTO 2420
3050
3060 *****
3070 *
3080 * Copyright by T E K M A T E INC.
3090 *
3100 *****
3110 *****
3120 END

```



## パソコン用1次元イメージ・センサ・ボードの設計・製作



白川光英

工業用の測定、検査にはライン・センサ・カメラが広く利用されています。ここでは、このカメラをパソコン上で利用するためのインターフェース・ボードの実例について紹介します。

イメージ・センサというと、現在では家庭用小型ビデオ・カメラに使われている2次元のCCDイメージ・センサを思い起こされる人が多いと思われます。しかし、第14章でも紹介するように、**イメージ・センサには1次元のものと2次元のもの**とがあります。

確かに2次元CCDイメージ・センサが、高速に、また画素数がより多くなるに伴い、1次元のイメージ・センサは2次元のものにとって変わる用途も少なくないと思います。しかし、**1次元のイメージ・センサは、コスト面と高分解能を買われ、目立たない所ではずいぶん利用されています。**

よく知られているものでは、ファクシミリに使用され、文字や図形、絵を読み取って、情報の伝送に大きな役割を果たしています。また、昨今脚光をあびたオート・フォーカス・カメラの位相差検出による自動焦点機構にも応用されています。

このように、1次元のイメージ・センサは、高速読み取りを必要とする所には高速応答のものを、精度を必要とする所には画素数の多いものを、コストを追求される所にはカスタム化を、と今後もまだまだ多方面での利用が期待されています。

というわけで、ここでは1次元イメージ・センサ用のパソコン・インターフェース・ボードを製作し、測定/検査への応用を考えてみることにします。

### 1次元イメージ・センサ・カメラについて

#### ● 近ごろでは

イメージ・センサは、民生用だけでなく産業用とし

ての利用も年々増加しています。この分野においても、2次元イメージ・センサの利用はかなり進んできており、制御面からも各社より画像処理機器が発売されています。そして、パソコンに画像データを取り込んだり、あらかじめ作られたソフトを呼び出し、**被測定物の面積比較や重心検出を、パソコンで行える**ボードやコントローラが多く発売されています。

一方、1次元イメージ・センサの制御は、各センサ・メーカーが、自社のイメージ・センサ・カメラと同時に各種のコントローラを発売しています。しかし、それらのほとんどは、イメージ・センサのビデオ信号をいったんコントローラ内に取り込み、スライス・レベルで比較した後、データとしてRS-232Cなどのシリアル伝送や、BCDやバイナリ符号出力の形で、パソコンやシーケンサへ受け渡しを行っています。

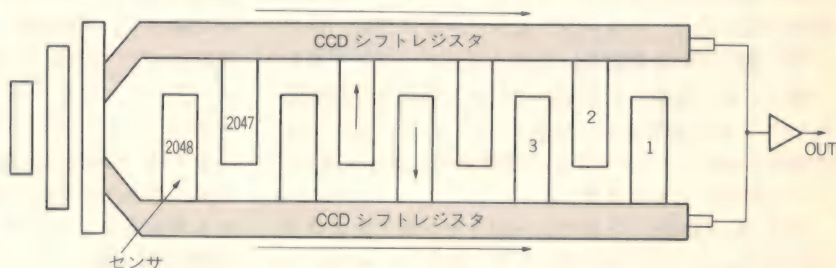
ところで、実際にイメージ・センサを利用したいという話を聞くと、高速で処理する必要性、たとえば**プリント基板の孔を高速で数えたり、落下する部品の数を数えたり**と言う処理は、ソフトでの対応にはスピード的に無理があります。

しかし、データをフロッピーに保存したい、データをプリント・アウトしたい、偏差値をグラフ化したい、というデータを加工する必要がある処理には、パソコンによるソフトウェアの利用は大変効果的です。

#### ● イメージ・センサの原理

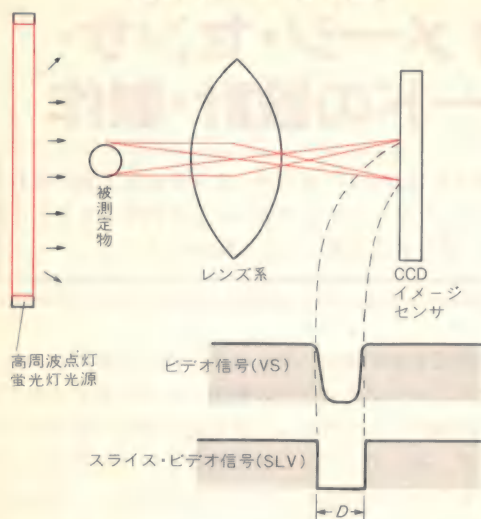
イメージ・センサ・カメラに使われているCCD (Charge Coupled Device=電荷結合素子) は、図1のように例えばフォト・ダイオードが2048個(2048ビットという)、あるいは4096個(4096ビットという)一列

〈図1〉  
1次元CCDイメージ・センサ  
基本構造





〈図2〉 イメージ・センサ原理



に並んで、その両側に二つのCCDシフトレジスタがトランスファ・ゲートを介して並んでいます。これがCCDの基本構造です。

この感光部(フォト・ダイオード)に入った光は、光電変換により、入光量に比例した電荷を蓄積します。そして、この電荷は両端に配置されたシフトレジスタにより次々と転送され、最終的にはアンプされて出力部から電気信号として取り出せます。

#### ● 1次元イメージ・センサ・カメラの利用法

イメージ・センサ・カメラは、このようなイメージ・センサ素子に、**レンズ系を通して被計測物をCCD上に結像させることにより取り出される電気信号を、光源に応じたスライス・レベルで比較して、制御の容易なON/OFFのデジタル信号に整形して、計測などに利用されます。**

例えば、円柱状の被測定物の直径を計測する場合は、図2のようにビデオ信号をカメラから取り出し、それを適当に設定したスライス・レベルで比較することにより、**スライス・ビデオ信号**を取り出すことができます。

そしてスライス・ビデオ信号の**時間幅Dのビット数をカウントして、1ビット当たりの分解能を乗算して**実際の被測定物の径を求めることができます。測定分解能に関しては、イメージ・センサ・カメラに、レンズ系を通して映る視野を、そのイメージ・センサのもつ素子の数で除算した値が、1ビット当たりの分解能ということになります。つまり、**[1ビット当たりの分解能(mm/ビット)=カメラ視野(mm)/イメージ・センサ素子数]**となります。

例えば、4096ビットのイメージ・センサ・カメラで、視野50mmのレンズ系を使つたとすると、イメージ・

〈表1〉 イメージ・センサ・カメラの素子性能

イメージ・センサ カメラ型式名	IX-S5	IX-S20	IX-S40
素子数	512ビット	2048ビット	4096ビット
素子の大きさ	25 $\mu$ m×25 $\mu$ m	14 $\mu$ m×14 $\mu$ m	7 $\mu$ m×7 $\mu$ m
素子の感度 不均一性	±5% (標準), ±10% (最大)		
走査周期	1 ms	1 ms	2 ms

センサ4096ビット中の1ビットの分解能は、50mmを4096で除算した値 0.012(12 $\mu$ m)となります。ただし、センサ素子の分解能は、メーカの資料によると、**素子数4096ビットのもので7 $\mu$ m×7 $\mu$ m**となっていますので、7 $\mu$ m以下の測定は絶対値ではなく相対値になります。

代表的な1次元イメージ・センサ・カメラの例として、サックス(株)のカメラ型式と素子の関係を表1に示します。

#### ● 1次元イメージ・センサ・カメラの使い方

イメージ・センサ・カメラについては、各社よりそれぞれ用途に合ったものが発売されていますが、計測用途に使用するにあたっては、分解能を上げるために**接写リング**を利用したり、**マクロ・レンズ**を利用したりと、その測定物に合ったレンズ選択が必要となります。

また、ラフな検出にはあまり問題にはならないレンズの明るさや、レンズの収差もシビヤな計測では重大な問題になります。こうしたことから、計測用途のカメラに関しては、市販の製品群の多いカメラ用レンズを使えるもののほうが有利です。

また、計測システムを考えるうえで、レンズ系と同様に選択に迷うのは、光源です。現在、最も多く利用されているのは**高周波電源で点灯させた蛍光灯光源**です。

そのほか、光量の多いハロゲン・ランプ光源、光軸の狭いガス・レーザ光源、狭い場所へも取り付けの可能な光ファイバ式光源などがよく使われています。

#### ● カメラ IX-S20について

ここでは、イメージ・センサ・カメラにサックス(株)製のIX-Sシリーズの中のIX-S20(**2048ビット**)のカメラを使って、NEC製のパーソナル・コンピュータ、**PC98シリーズにイメージ・センサ・データを取り込むインターフェース・ボード**を製作します。**写真1**がカメラとインターフェース・ボードの外観です。

まずIX-S20について説明します。図3がその構成です。このカメラは、カメラ本体に外部から走査信号を入力することにより、1回だけスキャンするEXT(外部同期走査)モードと、カメラ内部で自動的に走査



## イメージ・センサの利用方法

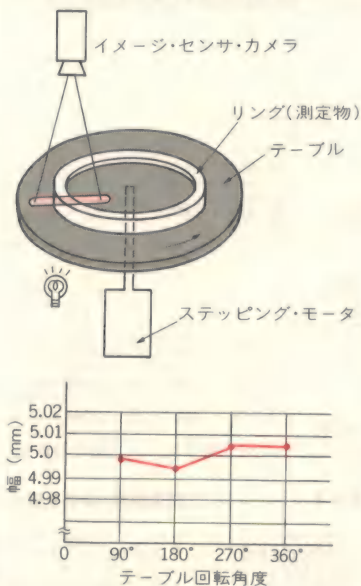
イメージ・センサ・カメラと今回のインターフェース・ボードを使ったシステムとして、次のような使い方があります。

### ● リング状の物の幅測定

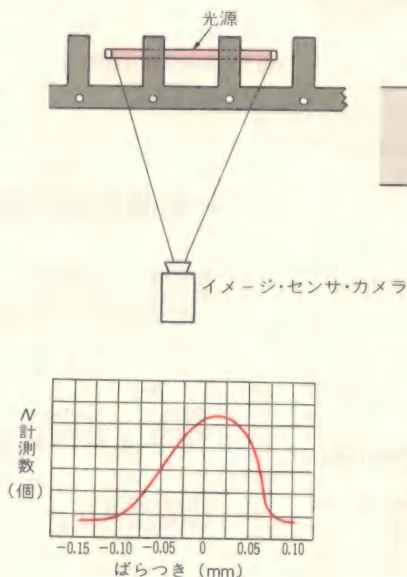
リング状の被測定物を、パルス・モータで回転できるようにしておきます。そして、パルス・モータを市販のPC98スロットに挿入できるパルス・モータ・コントローラで一定角度ずつ回転させます。この時のリング幅をイメージ・センサ・カメラで読み込み、実測データ値(mm)に変換して、測定角度-リング幅のグラフをプリント・アウトします(図A)。

### ● 製品のピッチ間計測

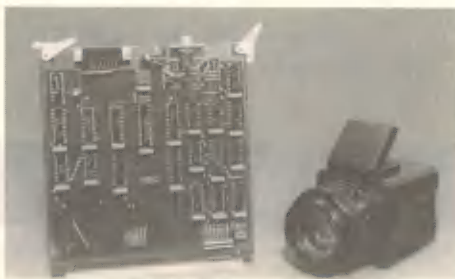
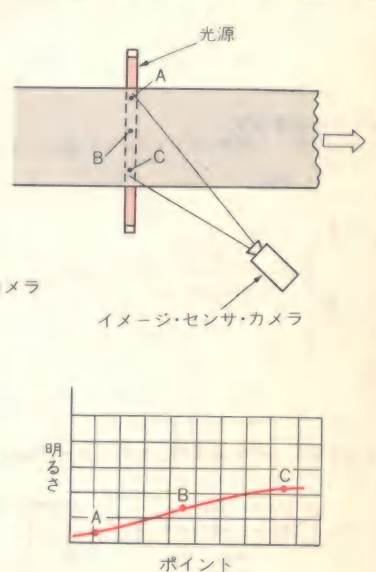
〈図A〉 リング状の物の幅測定



〈図B〉 プレス部品のピッチ間計測



〈図C〉 透明ボカシ・フィルムの明暗チェック



〈写真1〉 1次元イメージ・センサ・カメラとパソコン・インターフェース・ボード

信号を発生させ、連続的にスキャンするINT(内部自動走査)モードとをもっており、スイッチで切り替えることができます。

〈図3〉 カメラ本体図

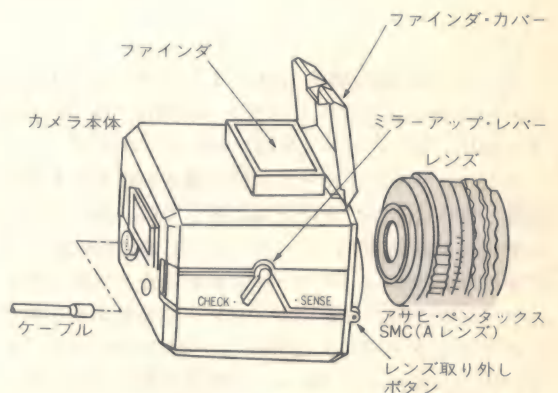




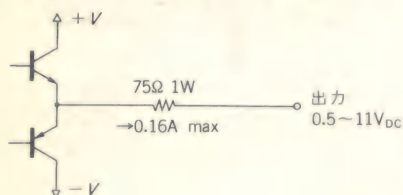
Diagram of the rear panel of the device showing various controls and connectors:

- スキャン・レート 切り替えスイッチ (Scan Rate Switch)
- 外部同期/自動走査 切り替えスイッチ (External Sync/Auto Search Switch)
- モニタ・ピン 上から VS, SP, GND (Monitor Pin: VS, SP, GND from top)
- 電源表示灯 (Power Indicator Light)
- 信号入出力コネクタ (Signal Input/Output Connector)

The diagram illustrates the CPU bus system for the TMS320C49 DSP. A central horizontal line represents the CPU bus, with labels for Control (コントロール), Address (アドレス), and Data (データ). Various components are connected to this bus:

- 機能選択 (Function Selection):** Receives control and address signals.
- レジスタ (Registers):** Connected to the data bus.
- コンバータ (Converter):** Receives data from the registers and outputs a signal labeled "画像信号" (Image Signal).
- VRAM (Video Random Access Memory):** Connected to the data bus.
- 読み出しカウンタ (Read Counter):** Receives address signals and outputs to the data bus.
- 書き込みカウンタ (Write Counter):** Receives address signals and outputs to the data bus.
- パルスコントローラ (Pulse Controller):** Receives address signals and outputs to the data bus.
- D-Aコンバータ (D-to-A Converter):** Receives data from the bus and outputs to the I/O block.
- I/O (Input/Output):** Connected to the D-A converter and the external world (外部へ).
- スレッシュホルド設定 (Threshold Setting):** Receives data from the bus and outputs to the I/O block.
- 同期信号 (Synchronization Signal):** Receives data from the bus and outputs to the I/O block.

受光信号，1画素(ビット)ごとに連続的に出力される。




The diagram shows three signals over time. The top signal, CP, is a periodic square wave. The middle signal, SP, is a square wave with a period consisting of a high phase (duration  $t_1$ ) and a low phase (duration  $t_2$ ). The bottom signal, VS, shows a ramp-up during the SP high phase and a flat top during the SP low phase. Vertical dashed lines align the transitions of SP with the corresponding changes in VS.

	$t$
IX-S5	$1\mu s$
IX-S20 IX-S40	$0.25\mu s$

	$t_1$	$t_2$
IX-S5	1024	$18 + (1024 \times n)$
IX-S20	1024	$24 + (1024 \times n)$
IX-S40	2048	$15 + (2048 \times n)$

 $n$ : 切り替えボリュームの目盛り

$t_2$  の間に走査し,  
 $t_1$  の間に VS が出る.

	目盛り	0 1 2 ... A B C D E F
	$n$	0 1 2 ... 10 11 12 13 14 15

スキャン・レート調整ボリュームで切り替える。

ただし、スキャン・レート変更はモード切り替えスイッチを「INT」側にした時に有効。

カメラの外部同期入力は、“L” アクティブ(エッジ・トリガ)、入力インピーダンス $470\Omega$ 、“H” レベル 3~12V、“L” レベル 1V以下となっています。

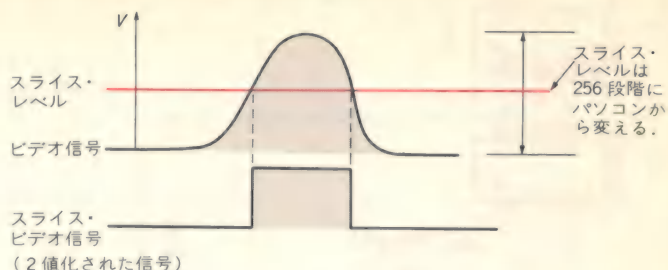
また、スキャン・レートの切り替えも図4のように装備しています。さらに出力に関しては、カメラからの受光による電荷を、1ビットごとに連続的に出すビデオ信号(VS)、イメージ・センサを走査させるスキャン・パルス(SP)、イメージ・センサの基本クロックであるクロック・パルス(CP)、などが出力されています。出力、スキャン・レートとの関係を図5に示します。

イメージ・センサのインターフェース・ボードのハードウェアは図6のブロック図のようになっています。

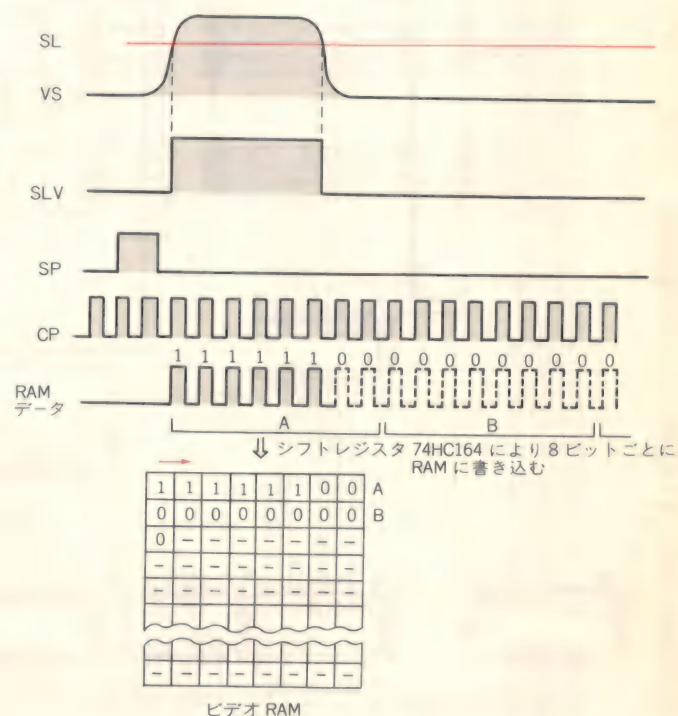
パソコンから出された8ビットのスライス・レベルのデータは、D-AコンバータAD558JDにより、0～2.56Vのアナログ電圧に変換されます。この電圧信号は、図7のようにカメラから入ってきた画像ビデオ



〈図7〉ビデオ信号とスライス・レベルとの関係



〈図8〉  
ビデオ信号をビデオRAMに取り込むまでの  
タイムチャート



信号(VS)とLM311(電圧コンパレータ)で比較され、2値化されます。

そして、2値化された信号は、イメージ・センサのクロック・パルス(CP)によりシフトされ(74HC164の8ビット・シフトレジスタ)、ビデオRAM TMM2015 BP12(2K×8ビット)に書き込みます。

### ● ビデオRAMの働き

このビデオRAMは、通常データ・バスとアドレス・バスが、パソコン側につながっています。このボードでは、2K×8ビットのビデオRAMをボード上にもっています。これは、ビデオ信号をカメラのクロックで最小時間に読み込むには、RAMをパソコン外にもち、そのRAMを読むほうがよいと考えたからです。また、8ビット単位で読み書きできるようにしたので、ソフトも容易になります。容量的には少し大きすぎるかもしれませんが、イメージ・センサの将来の発展を考え、RAMには2K×8ビットのものを使いました。

### ● 動作原理の概要

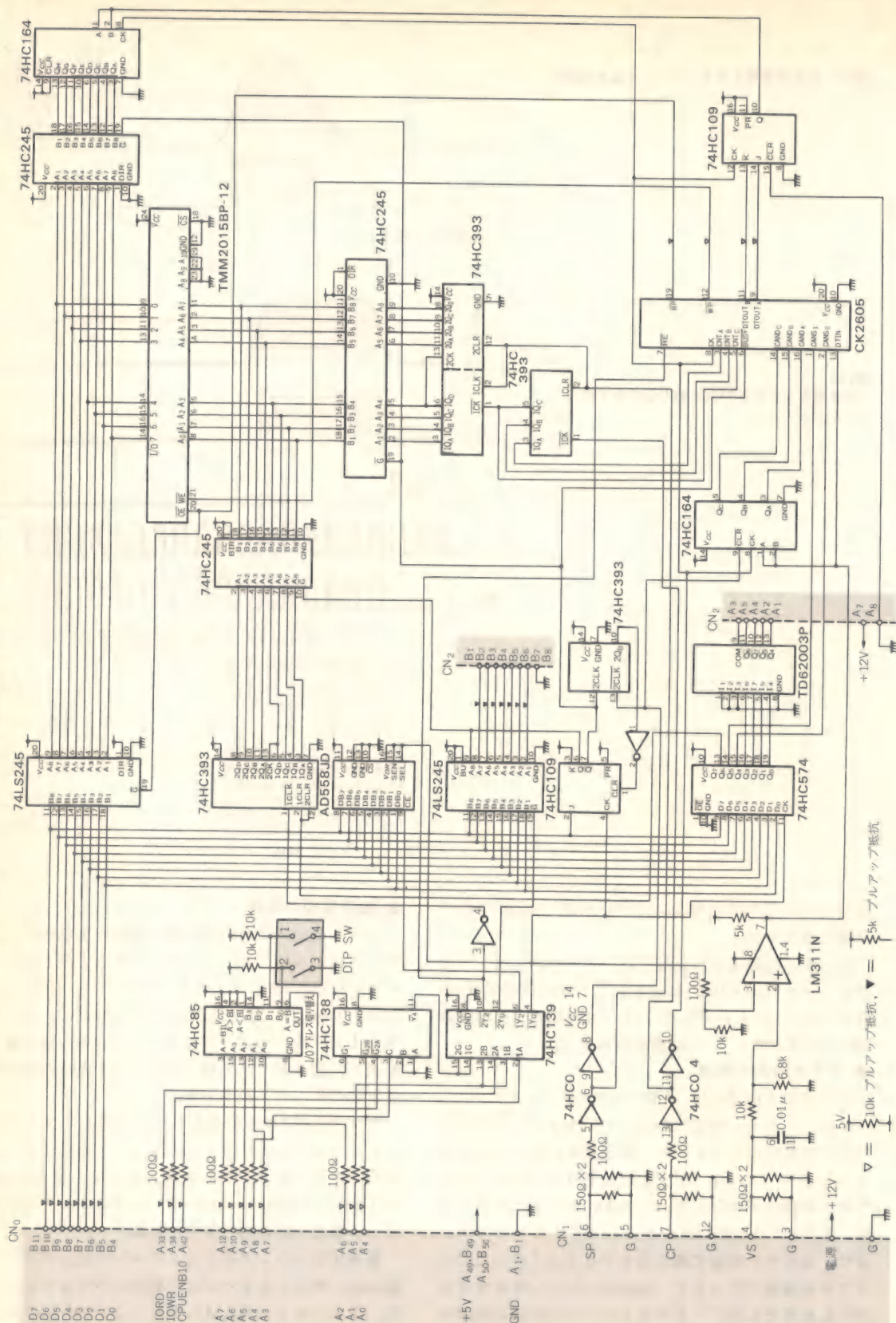
ソフトウェアについては後で説明しますが、ソフトウェアによってスタート・ビットを立てるにより、ビデオRAMのデータ・バスとアドレス・バスをカメラ側に切り替えます。そして、イメージ・センサ・カメラの1スキャン分のデータを、ビデオRAMに書き込みます。図8がビデオ信号をビデオRAMに取り込むときのタイムチャートです。

ビデオRAMからの画像データを読み出すときは、ソフトウェアにより、クリアのビットを立てて、読み出しカウンタをクリアします。そして、ビデオRAMのアドレスを0にしておいて、パソコンへデータを8ビット単位で読み込むようにします。

読み出しカウンタのインクリメントは、パソコンの読み出し信号があるたびに自動的に行うようにしました。なお、ビデオRAMへのリード/ライトの切り替えは回路を簡素化するためにPLA(プログラマブル・

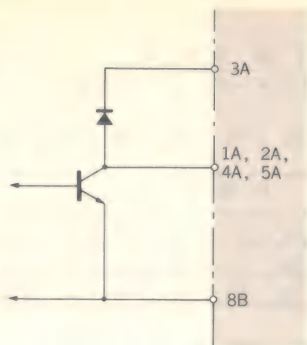


図9-40 イメージ・センサ・カメラ・インターフェース・ボード全回路図





〈図10〉 外部出力



〈表2〉 電源、入出力コネクタ表

ピンNo.	信号名	説 明
1 A	外部出力	外部出力4ビット中の第0ビット
2 A	外部出力	外部出力4ビット中の第1ビット
3 A	外部出力	図10参照
4 A	外部出力	外部出力4ビット中の第2ビット
5 A	外部出力	外部出力4ビット中の第3ビット
6 A	—	未使用
7 A	12V	イメージ・センサ用DC12V電源入力
8 A	0 V	(5 V系のグラウンドと共通)
1 B	外部入力	外部入力7ビット中の第6ビット
2 B	外部入力	外部入力7ビット中の第5ビット
3 B	外部入力	外部入力7ビット中の第4ビット
4 B	外部入力	外部入力7ビット中の第3ビット
5 B	外部入力	外部入力7ビット中の第2ビット
6 B	外部入力	外部入力7ビット中の第1ビット
7 B	外部入力	外部入力7ビット中の第0ビット
8 B	0 V	外部入力用グラウンド (5 V系のグラウンドと共通)

ロジック・アレイ)を使用しています。

また、実際に測定を行うとき、測定治具や機械側コントローラのやり取りに利用できるように、外部入力にはTTLレベル、外部出力にはTD62003P(トランジスタ・アレイ)を使い、50V、400mAのトランジスタ・オープン・コレクタ出力となっています。図9にインターフェース・ボードの全回路図、図10に外部出力、表2に電源、入出力コネクタ接続表を示します。

#### ● 使用するための準備

イメージ・センサ信号入力ケーブルに関しては、表3にあるように、サンクス㈱の両側コネクタ付きケーブル(IX-CC 5 W)を使用する場合は、そのまま使えるようにコネクタのピンは同じ配列になっています。

電線関係は自作する場合には、電源線は0.18mm<sup>2</sup>以上1芯キャプタイヤ・ケーブル、968Ω/km以下のものを2本、信号ケーブルは0.06mm<sup>2</sup>以上の特性インピーダンス75Ωの同軸ケーブル4本(717Ω/km以下)を使用します。

入出力および電源用コネクタにはヒロセ電機のHR10-10P12Sを使ってください。外部入出力ケーブルは0.18mm<sup>2</sup>以上の線を使用します。

カメラ信号用のコネクタには、富士通㈱製の

〈表3〉 イメージ・センサ信号入力コネクタ表

ピンNo.	信号名	説 明
1	0 V	イメージ・センサ用DC12V電源出力
2	12V	
3	0 V	ビデオ・シグナル・グラウンド
4	VS	ビデオ・シグナル
5	OV	スタート・パルス・グラウンド
6	SP	スタート・パルス
7	CP	クロック・パルス
12	OV	クロック・パルス・グラウンド

〈表4〉 I/Oアドレス表

DIP SW No.		I/Oアドレス	
1	2	1	2
ON	ON	D <sub>0</sub>	D <sub>2</sub>
ON	OFF	D <sub>4</sub>	D <sub>6</sub>
OFF	ON	D <sub>8</sub>	DA
OFF	OFF	DC	DE

〈表5〉 使用準備のまとめ

- (1) ボードのI/Oアドレスを設定する。
- (2) ボードをスロットに実装する。
- (3) イメージ・センサ信号ケーブルと外部入出力ケーブルを接続する。
- (4) イメージ・センサ・カメラのモード切り替えスイッチをINT<sub>1</sub>にセットする。(図4参照)
- (5) イメージ・センサ・カメラのスキャン・レートの切り替えは0にセットする。(図4参照)
- (6) 光源を点灯する。
- (7) イメージ・センサの電源(DC12V)をONする。
- (8) イメージ・センサ・カメラのピント絞りを調整する。
- (9) パソコンを起動する。

FCN365J016AUを使ってください。

また、このボードのI/Oアドレスは、ディップ・スイッチNo.1、No.2により設定します。表4にディップ・スイッチとI/Oアドレスの関係を示します。パソコンの機種によっては、ユーザに開放されていないスロットがありますので注意が必要です。詳細はパソコンの説明書を見て確認してください。

表5に、実際にセットアップするための項目をまとめて示します。

なお、ピントと絞りの調整については、カメラIX-S20にはビデオ信号(VS)、スキャン・パルス(SP)、GNDのチェック・ピンが出ていますので、オシロスコープをつないで、調整が行えます。特に明暗のはっきりしない被測定物や、小さな被測定物には、この調整が必要となります。

## ソフトウェアについて

### ● I/Oアドレスの割り当て

I/Oの割り当てとそのビット割り当ては、表6のようになっています。このうちパラメータは、視野内に入ってくる被測定物の汚れやゴミ、ピンホールなどと



〈表6〉 I/O割付表

I/Oアドレス	I/O	機 能	ビ ッ ト 割 り 付 け							
1	O	スタート クリア パラメータ 外部出力	スタート	クリア	パラメータ		外部出力			
					D <sub>1</sub>	D <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
	I	ビジィ 外部入力	ビジィ	外 部 出 力						
2	O	スライス・レベル	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
	I	ビデオ・データ出力	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>

〈表7〉

キャンセル・ビット  
設定表

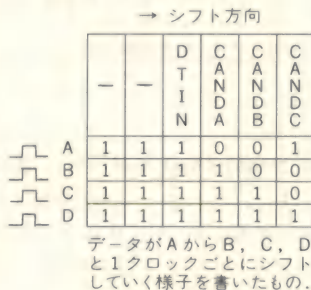
キャンセル ・ビット数	デー タ	
	D <sub>1</sub>	D <sub>0</sub>
0	0	0
1	0	1
2	1	0
3	1	1

〈図11〉 キャンセル・ビットの使い方

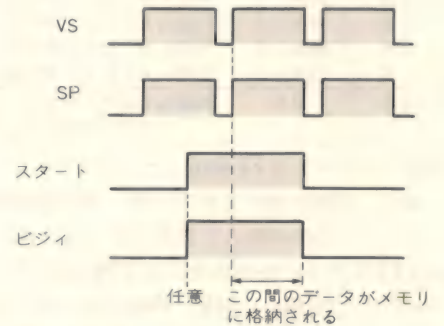
2 値化したデータ	1 1 1 1	0 0 0 1	1 1 1 1	0 0 1 1	1 1 0 1	1 1 1 1
キャンセル・ビット数を0 (D <sub>0</sub> =0, D <sub>1</sub> =0) にしたとき	1 1 1 1	0 0 0 1	1 1 1 1	0 0 1 1	1 1 0 1	1 1 1 1
キャンセル・ビット数を1 (D <sub>0</sub> =1, D <sub>1</sub> =0) にしたとき	1 1 1 1	0 0 0 1	1 1 1 1	0 0 1 1	1 1 * 1	1 1 1 1
キャンセル・ビット数を2 (D <sub>0</sub> =0, D <sub>1</sub> =1) にしたとき	1 1 1 1	0 0 0 1	1 1 1 1	* 1 1 1	1 * 1 1	1 1 1 1
キャンセル・ビット数を3 (D <sub>0</sub> =1, D <sub>1</sub> =1) にしたとき	1 1 1 1	* 1 1 1	1 1 * 1	1 1 1 *	1 1 1 1	1 1 1 1

\* : はキャンセル・ビット数の設定により、読み込みをスキップされた所。

〈図12〉  
キャンセル・ビットの  
しくみ



〈図13〉  
タイム  
チャート



被測定物とは区別するために、**設定値以下のビットを占めるものをキャンセルする機能**です。データとキャンセル・ビットの数の関係を表7に示します。

#### ● キャンセル・ビットについて

このキャンセル・ビットは、パソコン側からパラメータとして、D<sub>0</sub>, D<sub>1</sub>のいずれか、または両方を立てることにより、CK2605のロジック・アレイに、CANS<sub>0</sub>とCANS<sub>1</sub>として入力されます。この二つの入力判断より、図11のようにキャンセル・ビット数を設定します。そして、その設定値にしたがって、シフトレジスタ 74HC164により、シリアル信号をパラレル信号に変え、その3ビット；CANDA, CANDB, CANDC, を見ることににより、キャンセルすべき信号かどうかを判断します。この内容は図12を使って説明しましょう。

図12で、ロジック・アレイ CK2605の入力信号

CANDCがBの状態で、“1”から“0”に変化したと仮定します。すると、それに続くCANDB, CANDA, DTINに、CANDCと同符号の信号、この場合は“0”が、キャンセル・ビット数以上続かないと、次にCANDCの信号が変化するまでは、74HC164-2のシフトレジスタに信号を送らず、ビデオRAMに、データを書かないようになっています。

なお、これ以上のビットのキャンセルを考える場合は、ソフトで対応するようにします。

#### ● 動作させるには

スタートは1を書くことで、1スキャン分の画像データをビデオRAMに書き込みます。

クリアは1を書くと、ビデオRAMのアドレス・カウンタをクリアします。すなわち、ビデオRAMアドレスを0番地にします。この後、すぐにクリアのビットは0にもどしてください。



## 〈リスト1〉プログラムmemd

```

10 'save"2:memd"
20 CONSOLE 0,25,0,0:COLOR 0
30 CLS 3
50 '
60 Z = &H0
70 GOSUB *OUTDO
80 '
90 OUT &HD2,&H80: 'slice level set
100 X = ZZ AND &HCF
110 Z=X OR &H0: 'not cancel
120 GOSUB *OUTDO
130 LOCATE 5,0
140 PRINT "00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f"
150 PRINT
160 GOSUB *ADCL: 'adrs clear
170 GOSUB *SCAN: 'scan start
180 A=INP(&HD0) AND &H80: 'busy
190 IF A=128 THEN 180
200 D$=""
210 FOR J=0 TO 15
220   FOR I=0 TO 15
230     DAT=INP(&HD2)
240     DAT$=RIGHT$("0"+HEX$(DAT),2)+" "
250     D$=D$+DAT$
260   NEXT I
270   X$=HEX$(J)+"0"+" " +D$
280   PRINT X$
290   D$=""
300 NEXT J
310 GOTO 130
320 '
330 *ADCL
340 Z=ZZ OR &H40
350 GOSUB *OUTDO: 'set "1"
360 Z=ZZ AND &HBF
370 GOSUB *OUTDO: 'set "0"
380 RETURN
390 '
400 *SCAN
410 Z=ZZ OR &H80
420 GOSUB *OUTDO: 'set "1"
430 Z=ZZ AND &H7F
440 GOSUB *OUTDO: 'set "0"
450 RETURN
460 '
470 *OUTDO
480 OUT &HD0,Z
490 ZZ=Z
500 RETURN

```

ビジィは、ビデオ・カメラのスキャン中“1”がセットされますので、これが“0”になってからデータの読み出しを行います。図13にタイムチャートを示しています。

スライス・レベルは256段階に変化させることができます。値が大きくなるほどスライス・レベルは高くなります。ビデオ・データ出力は、スライス・レベルで2値データに変えられた画像のデータ出力で、1回のリードで8ビットのデータが読み出せます。このI/Oアドレスを1回リードすると、自動的にメモリの読み出しアドレスが一つ進みます。

### ● モニタ・プログラムの作成

ここでは、わかりやすくするためにBASICによるモニタ・プログラムを作成しました。

**プログラム名memd:**(リスト1)は読み取ったデータを1バイトごとにHEX形式で画面に表示するプログラムです。

90:OUT &HD2, &H80で出力アドレスD<sub>2</sub>にH80を出力して、スライス・レベルを中心にセットしています。

330:\*ADCLのサブルーチンでは、ビデオRAMアドレスのクリアを行っています。つまり、クリアのビットを立てた値H40を出力しています。そして、すぐにクリア・ビットを0にもどしています。

## 〈リスト2〉プログラムvm

```

1000 'save "vm"
1010 'video monitor
1020 '
1030 CONSOLE 0,25,0,0:COLOR 0
1040 SCREEN 2,0,0,1
1050 CLS 3
1060 '
1070 Z=&H0
1080 GOSUB *OUTDO
1090 '
1100 DIM V(512)
1110 FOR I=0 TO 511:V(I)=0:NEXT I
1120 FOR SL=0 TO 255 STEP 32:LOCATE 1,1:PRINT SL
1130   OUT &HD2,SL: 'slice level set
1140   GOSUB *SCAN
1150   GOSUB *ADCL
1160   VBC=1
1170   FOR VDC=1 TO 256
1180     D=INP(&HD2)
1190     FOR BC=7 TO 3 STEP -4
1200       B=0
1210       IF D AND 2^BC THEN B=1: 'bit read
1220       IF B<>0 THEN *ALSET
1230       IF V(VBC)<>0 THEN *ALSET
1240       *ALSET
1250       VBC=VBC+1
1260     NEXT BC
1270     NEXT VDC
1280     NEXT SL
1290 '
1310 FOR I=1 TO 512
1320   IF V(I)=0 THEN V(I)=255
1330 NEXT I
1340 '
1350 CLS 3
1360 WX=50:WY=380
1370 FOR I=1 TO 512
1380   PSET (WX+1,WY-V(I))
1390 NEXT I
1400 END
1410 '
1420 *SCAN
1430 Z=ZZ OR &H80
1440 GOSUB *OUTDO
1450 Z=ZZ AND &H7F
1460 GOSUB *OUTDO
1470 RETURN
1480 '
1490 *ADCL
1500 Z=ZZ OR &H40
1510 GOSUB *OUTDO
1520 Z=ZZ AND &HBF
1530 GOSUB *OUTDO
1540 RETURN
1550 '
1560 *OUTDO
1570 OUT &HD0,Z
1580 ZZ=Z
1590 RETURN

```

400:\*SCANのサブルーチンでは、スタートのビットを立てた値H80を出力して、その後スタート・ビットを0にもどして、画像データをビデオRAMに取り込んでいます。

**プログラム名vm:**(リスト2)は、実際にイメージ・センサ・カメラのレンズ系の調整などには、本来オシロスコープを使って調整するのですが、VS波形をオシロスコープで見たようにCRTに出力するプログラムです。

ここでは、スライス・レベルを32ステップごとに変化させて、PSET命令を使って描いています。

では、実際に先の図2で書いたような構成で、丸棒のような被測定物の直径に相当する、イメージ・センサのしゃ光部分のビット数を求めるプログラム名:**dsp811**(リスト3)を作ってみましょう。

まず、1050:~1080:ではキャンセル・ビットを入力するようにしてあります。そして、1120:~1130:ではビデオ信号の立ち上がり、また



### 〈リスト3〉プログラムdsp811

```

1000 'save "dsp811"
1010 CLS 3
1020 CONSOLE 0,25,0 0:COLOR 0:CLS 3
1030 Z=&H0
1040 GOSUB *OUTD0
1050 INPUT "cansel bit (0,1,2,3)":CB
1060 IF CB=0 THEN Z=&H0
1070 IF CB=1 THEN Z=&H10
1080 IF CB=2 THEN Z=&H20 ELSE Z=&H30
1090 GOSUB *OUTD0
1100 GOSUB *ADCL
1110 OUT &HD2,&H80:'slice level set
1120 INPUT "address1=":AD1
1130 INPUT "address2=":AD2
1140 PRINT "start%%%%hit any key"
1150 AS=INKEY$
1160 IF AS="" THEN 1150
1170 PRINT "start"
1180 GOSUB *SCAN
1190 A=INP(&HD0) AND &H80:'busy
1200 IF A=128 THEN 1190
1210 GOSUB *DISP3
1220 END
1230 '
1240 *DISP3
1250 EGD0=0:EGD=1:BIT=0
1260 WX=60:WY=60
1270 FOR K=1 TO 8
1280 FOR J=1 TO 32
1290 VD=INP(&HD2)
1300 T=VD
1310 FOR I=7 TO 0 STEP -1
1320 SD=0
1330 T=VD AND 2^I
1340 IF T=0 THEN 1360
1350 SD=1
1360 IF SD=0 THEN 1410
1370 PSET (WX,WY)
1380 EGD1=1
1390 IF EGD0=0 THEN EG=EG+1:EGD0=1 ELSE EGD0=1
1400 GOTO 1430
1410 EGD1=0
1420 IF EGD0=1 THEN EG=EG+1:EGD0=0 ELSE EGD0=0
1430 LOCATE 0,22:PRINT "All adrs number(s) = ";EG;
1440 IF EG>AD1 AND EG<AD2 THEN BIT=BIT+1:PRINT "A1 - A2 bit. numbre(s) = ";BIT
1450 WX=WX+1
1460 NEXT J
1470 AD=AD+1
1480 NEXT K
1490 WX=60
1500 WY=WY+10
1510 NEXT K
1520 RETURN
1530 *SCAN
1540 Z=ZZ OR &H80:'set "1"
1550 GOSUB *OUTD0
1560 Z=ZZ AND &H7F:'set "0"
1570 GOSUB *OUTD0
1580 RETURN
1590 *ADCL
1600 Z=ZZ OR &H40:'set "1"
1610 GOSUB *OUTD0
1620 Z=ZZ AND &HBF:'set "0"
1630 GOSUB *OUTD0
1640 RETURN
1650 *OUTD0
1660 OUT &HD0,Z
1670 ZZ=Z
1680 RETURN

```

は立ち下がりごとに、順次アドレスをふるようになっています。この場合、AD1=2、AD2=3(※1)を入力すると、棒の径に相当するビット数が求められます。

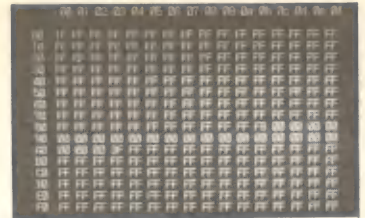
\*DISP3のサブルーチンでは、1290:で入力したビデオ信号に対して、1330:で各ビットのANDを取ることで、「0」、「1」の判別を行っています。そして、PSET命令により、スライス・レベルで比較したスライス・ビデオ波形を描くと同時に、指定したアドレス間のビット数を表示するようになっ

ています。

写真2に、それぞれのプログラムを実行して棒の径の測定を行っている時の様子を示します。

なお、このパソコン・インターフェース・ボードは(株)光伸舎(☎075-661-3161)より発売されています。

(注1)ここでアドレスを1から2に設定しないのは、ハードの関係上、最初の8ビットはLに落ちて、アドレスが発生するから。



(a) mend



(b) vm



(c) dsp 8 | 1

### 〈写真2〉プログラム実行例



## 2次元CCDカメラの設計・製作



長谷川孝美

2次元ビデオ・カメラ(いわゆる普通のビデオ・カメラ)も、CCDの低価格化と周辺LSIの発展により身近になってきました。ここでは、具体的なカメラの実例と、その活用のためのノウハウについて紹介します。

画像処理技術や、それを取り込んだロボット技術の発展には、視覚センサとしての固体撮像素子の開発が大きく寄与しています。また、この固体撮像素子はビデオ・カメラ、ファクシミリ、コピー・マシンなどの中にも見ることができ、急速に身近なものとなりつつあります。

そこで、ここでは一般に入手できる固体撮像素子〔インターライン型2次元CCD ICX018CL、ソニー(株)製〕を使った、2次元CCDカメラ(白黒)の製法について紹介することにします。このカメラは白黒ですが、カラー・テレビ放送の標準方式(NTSC)に準じていますので、一般のビデオ・カメラとして、あるいは画像処理システムの視覚センサとして、従来のITV(産業用テレビジョン)カメラと同様に利用することができます。

なお、ここで使用したCCDの表面にR、G、Bの原色フィルタを貼り付けたICX018CKは、カラー用CCDとして、8mmビデオ用カメラに使用されています。

### 撮像素子の動作

#### ● CCDイメージ・センサについて

テレビ用カメラの性能は、基本的には撮像デバイスによって決定されます。従来は撮像管(ビジコン管など)を使ったものが主流だったのですが、最近では固体撮像素子であるCCD(Charge Coupled Device)が多く使用されるようになりました。固体撮像素子は撮像管に比べると、**焼付きがないため寿命が半永久的**になる、という点が大きな特徴です。

固体撮像素子は最近ではイメージ・センサという呼び方もされており、用途およびしくみによって1次元イメージ・センサ、2次元イメージ・センサというように分類されています。

1次元イメージ・センサというのは、被写体が一定速度で移動したりするものの場合に有効なカメラで、イメージ・スキャナ、ファクシミリ、コピー・マシンなどに使用されています。ライン・センサ・カメラとも呼ばれているものです。

2次元イメージ・センサは、**被写体が静止していてもイメージをとらえることができる**もので、いわゆる一般的なビデオ・カメラに相当します。エリア・センサという呼び方もされています。これから紹介するのは、このエリア・センサを使った2次元のカメラです。まずは、撮像のためのエリア・センサの特性から紹介することにしましょう。

#### ● エリア・センサの種類

エリア・センサは、光を電気信号に変換する光電変換部と、その信号電荷を転送する方式によっていろいろの種類が考えられています。以下に3種類の代表的な特性について述べます。

##### ▶ フレーム・トランスファ型CCD

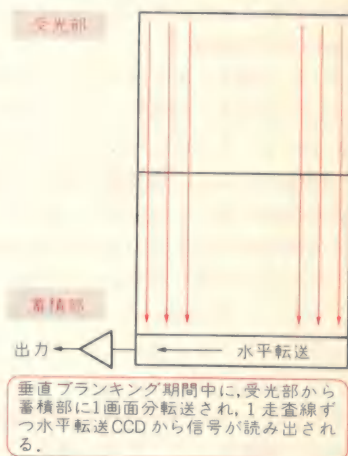
これは図1に示すように、光電変換を行う受光部と、信号として読み出す蓄積部とが分離して配置されています。この素子の特徴としては、**受光素子の有効面積が大きく**とれるため、センサ感度を大きくとれることと、構成がシンプルなので素子の高解像度化が容易になります。

逆に欠点としては、蓄積部を別に必要とするため、デバイスのチップ・サイズが大きくなり、素子の歩留りが低下します。

##### ▶ インターライン型CCD

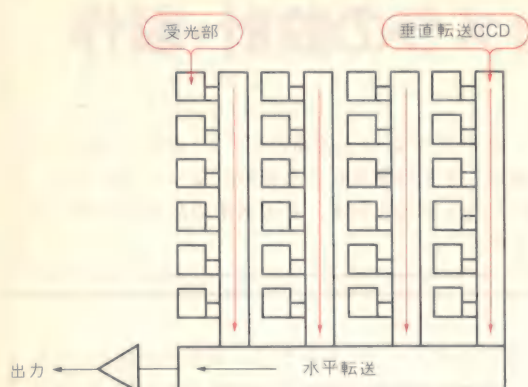
これは図2に示すように、受光部と転送部とが隣り

〈図1〉 フレーム・トランスファ型CCD

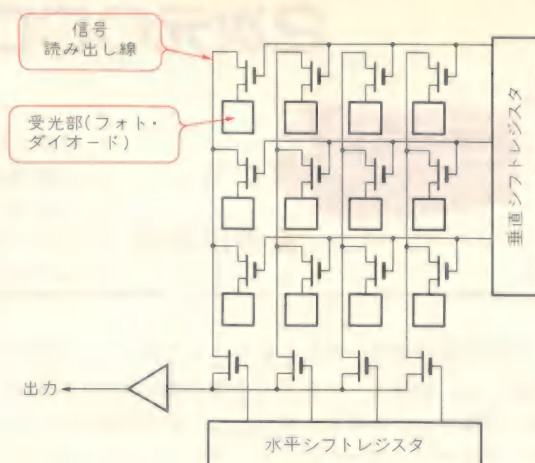




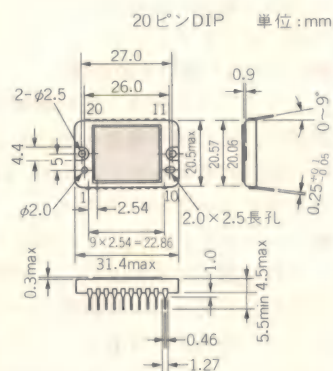
〈図2〉 インターライン型CCD



〈図3〉 MOS型撮像素子

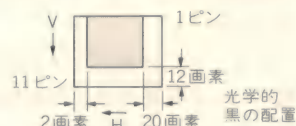


〈図4〉  
CCD ICX018CLの  
外形と特徴



(a) 外形寸法

- ・インターライン型CCDイメージ・センサ
  - ・ユニット・セル・サイズ  $17\mu\text{m}(\text{H}) \times 13\mu\text{m}(\text{V})$
  - ・ダミー・ビット数 水平8ビット、垂直1ビット(偶数フィールドのみ)
  - ・チップ・サイズ  $10.0\text{mm}(\text{H}) \times 9.3\text{mm}(\text{V})$
  - ・多重干渉効果を利用した薄膜多結晶シリコン・ゲートMOS型センサ
  - ・高感度出力増幅器内蔵
  - ・P基板Pウェル構造
  - ・有効画素数  $510(\text{H}) \times 492(\text{V})$
  - ・高感度低雑音
  - ・低スミア
  - ・アンチ・ブルーミング機能
  - ・低残像、焼き付きなし、図形ひずみなし、マイクロフォニック雑音なし
  - ・ $\gamma$ 特性: 1
  - ・光学的黒の数
- |          |    |      |
|----------|----|------|
| 水平(H) 方向 | 前部 | 2画素  |
|          | 後部 | 20画素 |
| 垂直(V) 方向 | 前部 | 12画素 |



(b) 素子の構造と特徴

合わせになるよう配置されています。特徴としては、**受光部から転送部への信号の転送が簡単**に行えることと、デバイスの**チップ・サイズが小さい**ことです。

反面、受光部と転送部とが隣り合わせの構造のため、素子の製作においてはミクロン・オーダの微細加工技術が必要になります。

#### ▶ MOS型撮像素子

これは図3に示すように、フォト・ダイオードで発生した信号を、MOSトランジスタによるX-Yスイッチ・マトリクスによって順次、出力に読み出します。

特徴としては、信号読み出し線が細くてすむため、受光部を大きくとれること、X-Yスイッチ・マトリクスの信号読み出しにいろいろな方法が考えられるので、**カラー化に対応しやすい**ことです。

ただし、欠点は、スイッチング時のノイズがCCDにくらべてとりにくいことです。

ここでは、以上のエリア・センサの中から一般に入手しやすい**インターライン型CCD**を使うことにして、カメラ設計に入ることにします。

## インターライン型CCD ICX018CL

ソニー(株)から発売されている1チップ白黒テレビ・カメラ用の、インターライン型CCD固体撮像素子ICX018CLについて説明します。

図4にICX018CLの外形寸法と特徴、素子構造を示します。このうち、主なトピックスは次のようなものです。

#### ● 有効画素数と光学的黒

有効画素数とは、受光部に配置されている受光素子のうち、被写体像を実際に光電変換する素子の数です。この**CCDでは水平方向に510個、垂直方向に492個**ありますので、全部で、

$$510 \times 492 = 250,920$$

となり、**約25万画素**になります。

これに対し、光学的黒と称する受光素子が、水平方向の前部2画素、後部20画素と垂直方向の前部12画素に配置されています。この受光素子には、光が入射し

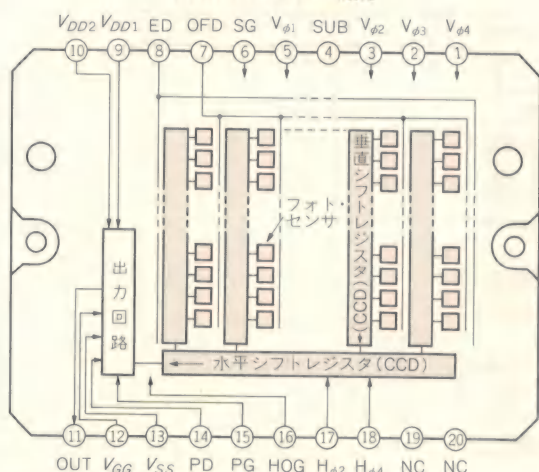


〈表1〉 ICX018CLの撮像特性

$T_a = 25^\circ\text{C}$

項 目	記号	最小	標準	最大	単位	備 考
感度	S	110	160	220	mV	
飽和信号	$V_{sat}$	400	530	850	mV	
映像信号シェーディング	$SV_i$		10	19	%	
スミア	SM		0.01	0.04	%	
暗信号	$V_{dt}$			13	mV	$T_a = 55^\circ\text{C}$
暗信号シェーディング	$\Delta V_{dt}$			4	mV	$T_a = 55^\circ\text{C}$

〈図5〉 ICX018CLの構成



(a) ブロック図

端子番号	記号	端子説明
1	$V_{\phi 4}$	垂直レジスタ転送クロック入力
2	$V_{\phi 3}$	垂直レジスタ転送クロック入力
3	$V_{\phi 2}$	垂直レジスタ転送クロック入力
4	SUB	基板
5	$V_{\phi 1}$	垂直レジスタ転送クロック入力
6	SG	センサ・ゲート・バイアス
7	OFD*	アンチ・ブルーミング・バイアス
8	ED*	エッジ・ドレイン・バイアス
9	$V_{DD1}$ *	電源電圧
10	$V_{DD2}$ *	電源電圧
11	OUT*	信号出力
12	$V_{GG}$ *	出力増幅器バイアス電位
13	$V_{SS}$ *	出力増幅器ソース電位
14	PD*	プリチャージ・ドレイン・バイアス
15	PG*	プリチャージ・クロック入力
16	HOG*	水平レジスタ読み出し制御バイアス
17	$H_{\phi 2}$	水平レジスタ転送クロック入力
18	$H_{\phi 1}$	水平レジスタ転送クロック入力
19	NC	無接続
20	NC	無接続

\*印の端子には負電位が印加されないよう注意が必要

(b) 端子配列

ないように素子の表面がしゃへいされていますので、得られる信号は**黒レベルの基準値**として使用されます。

### ● スミア

スミアとは、受光素子への入射光が結像位置以外にもれて垂直転送レジスタに入り込み、**明るい被写体が画面の垂直方向に尾をひく現象**をいいます。これは表1の撮像特性からわかるように、0.04%maxと小さく抑えられています。

### ● ブルーミング

ブルーミングとは、受光素子に**非常に明るい入射光があったとき、電荷があふれてまわりの受光素子に流れ込む**ことをいいますが、このCCDではブルーミングを生じないような対策が講じられています。

### ● $\gamma$ 特性

入射光に対する出力信号レベル特性をガンマといい、**入射光と出力信号レベルとが比例関係にあるとき $\gamma = 1$** と表現します。

### ● ユニット・セル・サイズ

受光素子と垂直転送レジスタによって構成される、1画素のサイズです。

有効受光部の寸法は、

$$17\mu\text{m} \times 510 = 8.67\text{mm (H)}$$

$$13\mu\text{m} \times 492 = 6.4\text{mm (V)}$$

となります。

### ● 動作のしくみ

図5にICX018CLのブロック図を示します。受光素子(フォト・センサ)の隣りに垂直転送レジスタがあり、垂直方向に電荷を転送します。

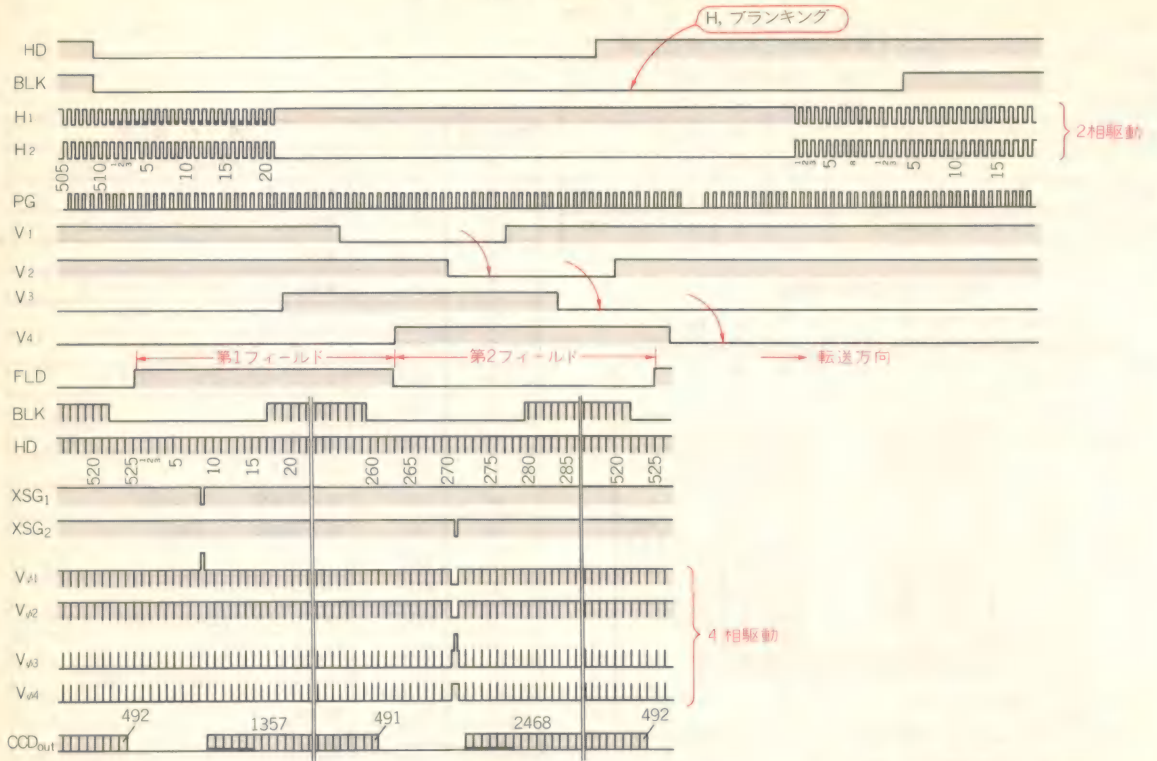
水平転送部まで送られた電荷は**1走査線ごとに水平転送レジスタによって転送**され、出力部から信号として読み出されます。図6にこのタイミングを示します。まず第1フィールドの受光素子に蓄えられた電荷は、XSG<sub>1</sub>のタイミングで垂直転送レジスタに転送されます。この電荷は垂直レジスタ転送クロック $V_{\phi 1} \sim V_{\phi 4}$ によって、 $V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4$ の方向に転送されていき、最後の $V_4$ の“H”，ブランキング中に、1走査線分の電荷が水平転送レジスタに転送されます。

水平転送レジスタに入った電荷は、水平レジスタ転送クロック $H_{\phi 1} \sim H_{\phi 2}$ によって $H_2 \rightarrow H_1 \rightarrow H_2 \rightarrow H_1$ の順に水平方向に転送されます。そして最後の $H_1$ からの電荷はPG(プリチャージ・クロック)によって読み出され、出力信号となります。

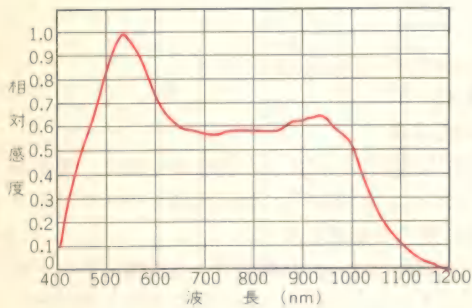
同様に第2フィールドの受光素子に蓄えられた電荷は、XSG<sub>2</sub>のタイミングで垂直転送レジスタに転送さ



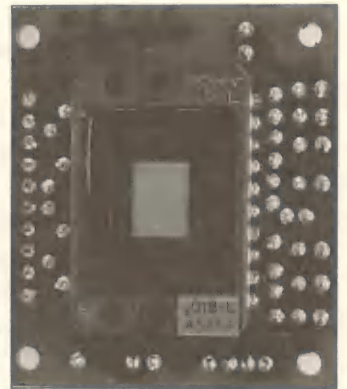
〈図6〉 ICX018CLの駆動タイムチャート



〈図7〉 ICX018CLの分光感度特性



〈写真1〉 CCD基板



れます。そして後は第1フィールドと同じ経路で読み出されます。

なおXSG<sub>1</sub>とXSG<sub>2</sub>は、2フィールドに1回しか出ませんので、このモードで動作するCCDを**フレーム蓄積モード**といいます。

#### ● 分光感度特性

図7にICX018CLの分光感度特性を示します。分光感度特性とは、光の波長に対する受光部の感度をいい、最も感度の良い波長を1.0にして相対値で表示します。このCCDは約1200nmまで感度がありますので、赤外線も撮ることができます(可視域は約400~700nm)。

写真1にCCDを基板上に実装したものを示します。

## CCDカメラの構成法

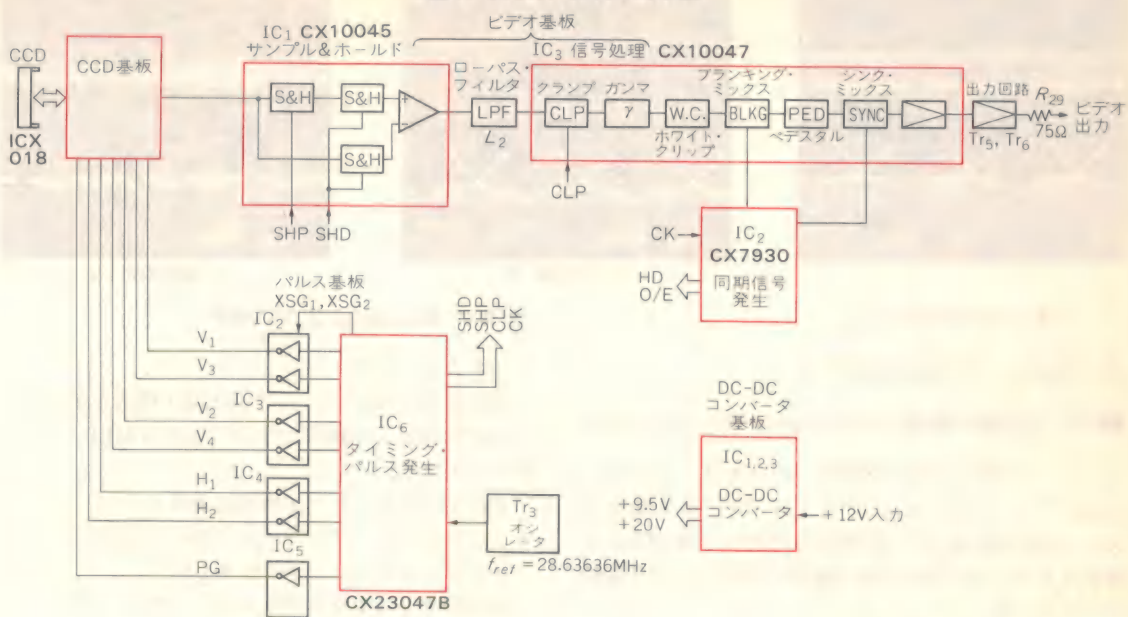
図8に今回試作した2次元カメラのブロック図を示します。表2が主な仕様です。回路基板構成としては、CCD基板、パルス基板、ビデオ基板、DC-DCコンバータ基板の構成です。以下、回路の構成について紹介します。

### ■ CCD回路

図9にCCD基板の回路構成を示します。CCD素子の電源電圧は20Vですが、出力部のバイアスとして、 $V_{PD}$ 、 $V_{SG}$ 、 $V_{GG}$ 、 $V_{HOG}$ の四つの電圧を供給します。



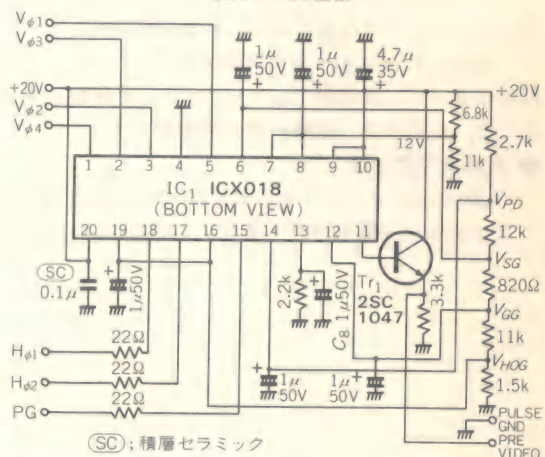
〈図8〉 CCDカメラのブロック図



〈表2〉 CCDカメラの主な仕様

撮像素子	インターライン転送方式CCD
有効画素数	510(H)×492(V)
撮像面積	8.7mm×6.4mm (2/3インチ撮像管相当)
水平周波数	15.734kHz
垂直周波数	59.94Hz
同期方式	内部同期
水平解像度	370 TV 本
垂直解像度	350 TV 本
最低被写体照度	15ルクス (レンズF1.4)
レンズ・マウント	Cマウント
フランジ・バック	17.526mm
走査方式	2:1 インタレース 525本 60フィールド/s
映像出力	1.0V <sub>P-P</sub> VBS 75Ω
電源入力	DC12V 約250mA

〈図9〉 CCD基板

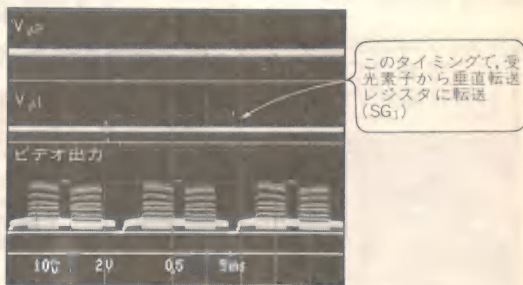


$V_{\phi 1} \sim V_{\phi 4}$ は垂直レジスタ転送クロックで、写真2のような波形をしています。

$V_{\phi 1}$ と $V_{\phi 3}$ は1フレームに1回、パルスの高レベルが約13Vまで上がり、その時に受光素子からの電荷が1フィールド分、垂直転送レジスタにいっせいに転送されます。

$H_{\phi 1}$ と $H_{\phi 2}$ は水平レジスタ転送クロックで、写真3のような波形をしています。水平転送レジスタの最終段まで送られた電荷は、プリチャージ・クロックPGのタイミングで、出力回路で電荷→電圧変換されます。そして、プリチャージ・パルスが重畳された信号として、11ピンより出力されます(写真4参照)。

この信号は、バッファ・トランジスタ $T_{R1}$ を経てビ

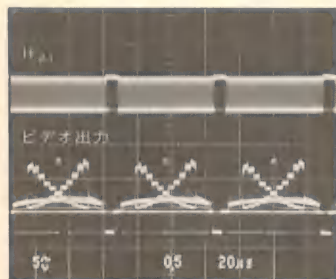


〈写真2〉ビデオ出力と $V_{\phi 1}$ ,  $V_{\phi 2}$

データ基板へ入力されます。

なお、水平転送レジスタ  $H_1$  と  $H_2$  は全部で540段になります。その内訳は、ダミー・ビット8段、光学的





〈写真3〉ビデオ出力と $H_{\phi 1}$

黒の数22段、有効画素510段です。

## ■パルス回路の構成

パルス回路の構成を図10に示します。これは、CCDドライバIC<sub>2</sub>~IC<sub>5</sub>、タイミング・パルス発生IC IC<sub>6</sub>、基準発振器Tr<sub>3</sub>、同期信号発生IC CX7930にて構成されます(CX7930は部品配置の関係でビデオ基板にのっている)。

## ●基準発振器の構成

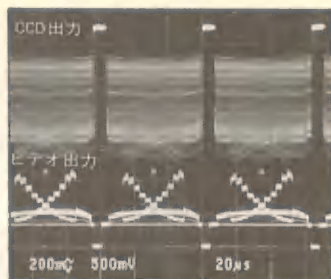
このCCDカメラ・システムは、カラー・カメラにも対応できるようになっており、発振周波数が、

$$8f_{sc} = 3.58 \times 8 = 28.63636\text{MHz}$$

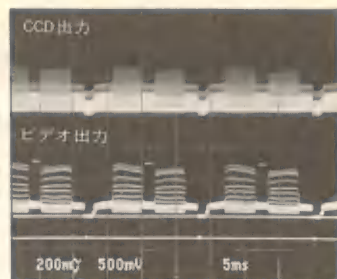
になるように、発振器Tr<sub>3</sub>のトリマVC<sub>1</sub>を調整します。ここで、 $f_{sc}$ はカラー副搬送波の周波数です。

## ●同期信号発生器

Tr<sub>3</sub>から出たクロックは、IC<sub>6</sub>で1/2に分周されて、ビデオ基板の同期信号発生器IC<sub>2</sub> CX7930にクロック



(a) 水平期間：H



(b) 垂直期間：V

〈写真4〉ビデオ出力とCCD出力信号

として入力します。

図11にCX7930のブロック図と端子構成を示します。このICは28ピンの構成ですが、外形はMFPという小型のものです。

このICからは、ビデオ信号処理用として、

- ・コンポジット・ブランキング信号 **BLKG**
- ・コンポジット・シンク信号 **SYNC**

CCDのタイミング・パルス用として、

- ・水平ドライブ信号 **HD**
- ・イーブン、オッド信号 **O/E**

を使用します。このイーブン、オッド信号とは、2：1インタレース方式のカメラにおいて、第1フィールドをオッド・フィールド(奇数フィールド)といい、第2フィールドをイーブン・フィールド(偶数フィールド)というのですが、それを示すための信号です。

図12にそれぞれのタイムチャートと、表3に日本が採用しているTV方式である、NTSC方式の規格の一

絶賛発売中!



CQ出版社

## ハードウェア・デザイン・シリーズ

★電子回路部品活用ハンドブックにつづく第2弾!

# アナログIC 活用ハンドブック

2色刷

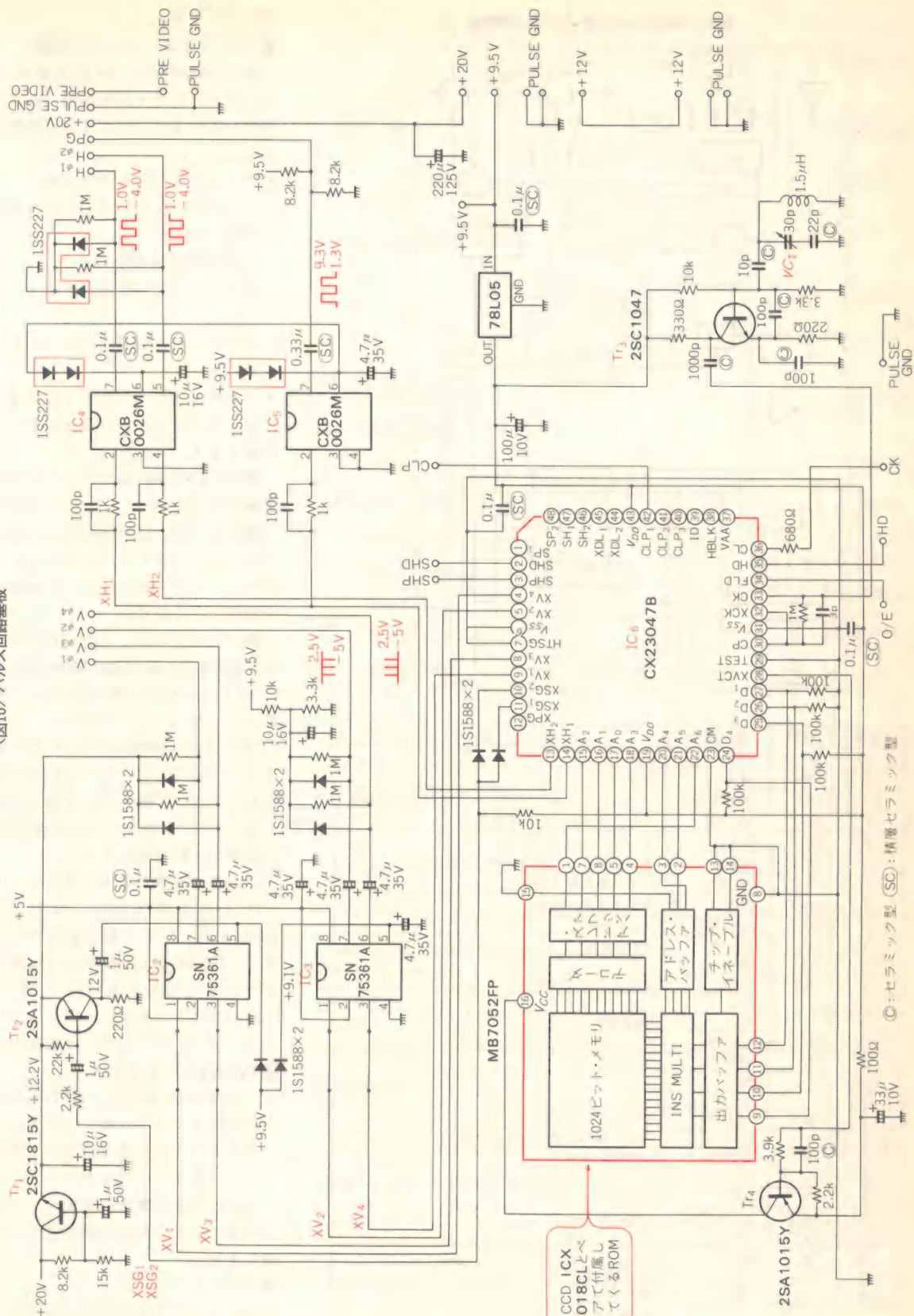
B5判 320頁  
定価 1,850円

内容

- |                |                     |
|----------------|---------------------|
| (1) プロローグ      | (6) 発振回路およびV-Fコンバータ |
| (2) OPアンプの基本機能 | (7) D-Aコンバータ        |
| (3) OPアンプと線形回路 | (8) A-Dコンバータ        |
| (4) 非線形の演算回路   | (9) 電源用IC           |
| (5) フィルタ回路     | (10) スイッチング・レギュレータ  |



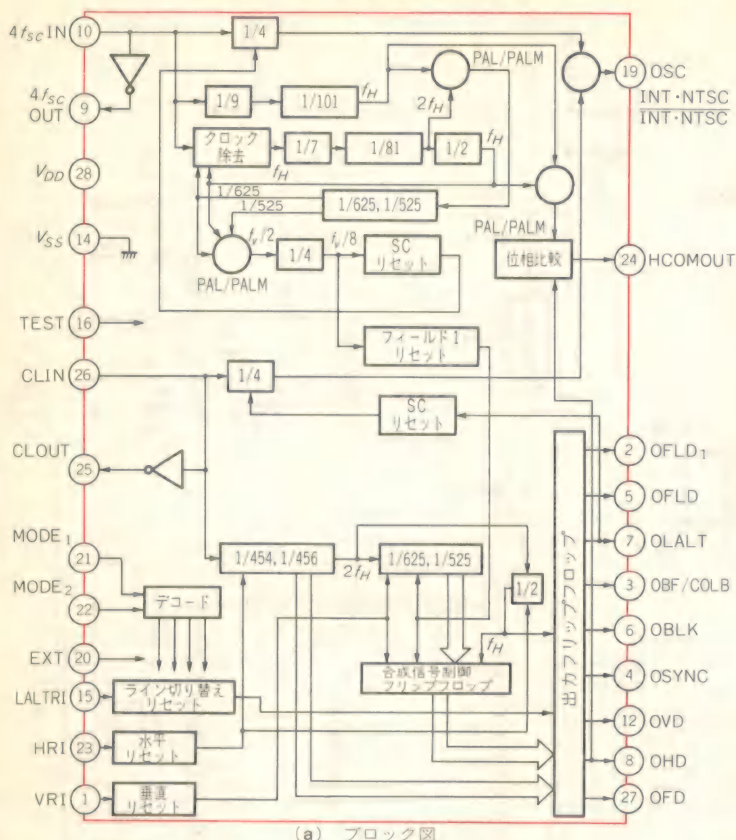
〈図10〉 パルス回路基板



◎：セラミック型 ◎SC：積層セラミック型



〈図11〉同期信号発生器 CX7930Aの構成



(a) ブロック図

端子番号	種類	端子記号	端子説明
1	IN	VRI	垂直リセット入力
2	OUT	OFLD <sub>1</sub>	第1フィールド出力
3	OUT	OBF/COLB	バースト・フラグ/カラー・ブランキング出力
4	OUT	OSYNC	コンポジット・シンク出力
5	OUT	OFLD	イーブン・オッド出力
6	OUT	OBLK	コンポジット・ブランキング出力
7	OUT	OLALT	ライン切り替え出力
8	OUT	OHD	水平ドライブ出力
9	OUT	4fsc OUT	4fsc 出力
10	IN	4fsc IN	4fsc 入力
11	—	NC	
12	OUT	OVD	垂直ドライブ出力
13	—	NC	
14	—	VSS	GND端子
15	IN	LALTRI	ライン切り替えリセット入力
16	IN	TEST	テスト入力
17	—	NC	
18	—	NC	
19	OUT	OSC	サブキャリア出力
20	IN	EXT	内部・外部同期モード切り替え
21	IN	MODE <sub>1</sub>	方式選択入力 1
22	IN	MODE <sub>2</sub>	方式選択入力 2
23	IN	HRI	水平リセット入力
24	OUT	HCOMOUT	位相比較器出力
25	OUT	CLOUT	クロック出力
26	IN	CLIN	クロック入力
27	OUT	OFH	水平周波数出力
28	—	VDD	電源端子

(b) 端子配列

部を示します。

## ● タイミング・パルス発生器

IC<sub>6</sub> CX23047は、CCDを駆動するパルスとビデオ信号処理パルスを発生しています。ビデオ信号処理用として、

- ・ クランプ・パルス信号 CLP
- ・ サンプル&ホールド・パルス SHP, SHD

CCDの駆動用として、

- ・ 垂直レジスタ転送クロック V<sub>1</sub> ~ V<sub>4</sub>
- ・ 水平レジスタ転送クロック H<sub>1</sub>, H<sub>2</sub>

- ・ プリチャージ・クロック信号 PG
  - ・ センサ・ゲート・クロック信号 SG
- があります。

図13にCX23047のブロック図と端子構成を示します。このICは48ピン構成ですが、QIP48と呼ぶ小型のフラット・パッケージに収められています。パルスのタイミングは図6を参照してください。

IC<sub>7</sub>のMB7052は、**CCDのキズ補正用ROM**(Read Only Memory)で、CCD ICX018CLを購入すると付属品としてついてくるものです。

CCDは25万画素の受光素子の集まりですので、どうしても製造工程においてキズの発生する可能性があります(キズが発生した受光素子のみ感度が大幅に変化する)。

このキズの数が少ない場合は、その素子を不良品とはしないで、ROMを用いてキズ補正動作を行います。ROMはCCDのキズの位置をデータとしてもちますので、CCDとはペアになります。

もちろんキズのないCCDにはROMは付いてきません。

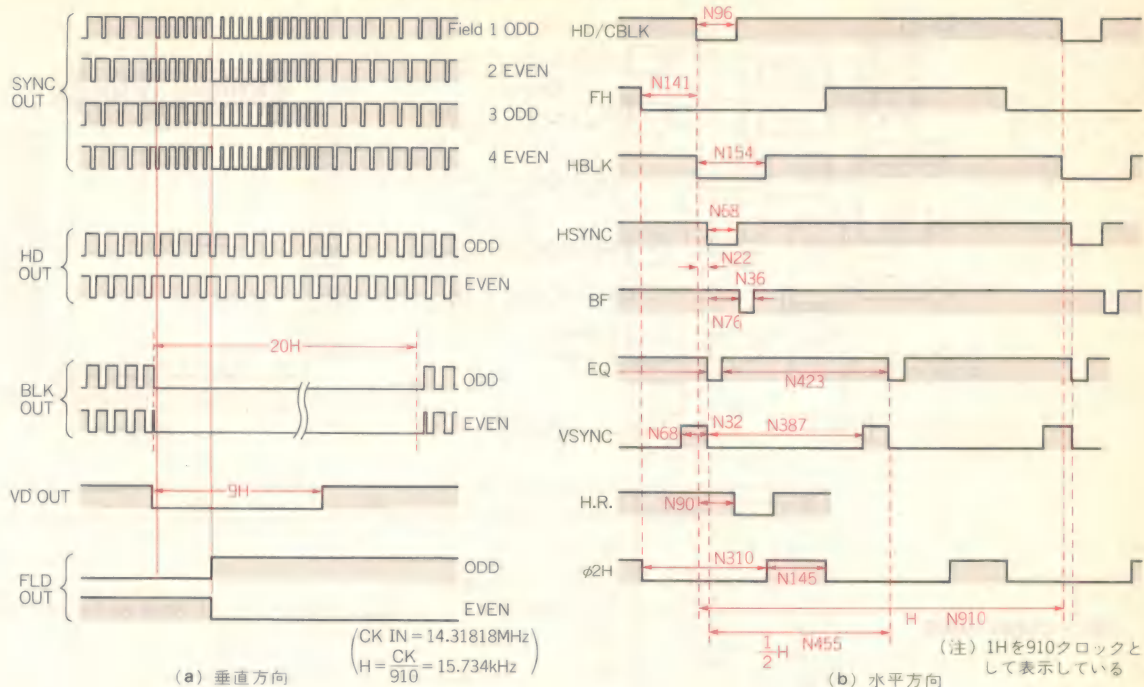
このROMの動作は、CCDからの出力信号をサンプル&ホールドしているとき、キズのある信号のときはサンプル&ホールドをストップして、一つ前の信号に置き換えます。すなわちキズのある部分は信号として使用しません。

## ● CCDドライバ

IC<sub>4</sub>, IC<sub>5</sub>(CXB0026)およびIC<sub>2</sub>,



〈図12〉 同期信号発生器 CX7930Aのタイムチャート



IC<sub>3</sub> (SN75361)は、5 V系パルスのレベル変換と、大容量負荷を高速でドライブするためのICです。図14にこのドライバICの構成を示します。CCD ICX018の垂直転送クロックとアース間の容量は、約6000pFの容量があります。そのため、このようなドライバICが必要になるのです。

また、ICX018のH<sub>φ1</sub>とH<sub>φ2</sub>、PGの周波数は、9.545 MHzで動作します。

Tr<sub>2</sub>はセンサ・ゲート・クロック SG<sub>1</sub>、SG<sub>2</sub>のレベルを高レベルに上げて、V<sub>φ1</sub>とV<sub>φ3</sub>を3値駆動させています。写真2にこの様子を示しています。また、写真3に水平転送クロックとビデオ出力との関係を示します。

## ■ビデオ信号処理回路

ビデオ信号処理回路の構成を図15に示します。この回路は、CCDからの出力信号をサンプル&ホールドするCX10045(IC<sub>1</sub>)と、クランプ、γ補正、ホワイト・クリップ、SYNCミックスなどを行うCX10047(IC<sub>3</sub>)で構成されます。図16がCX10045、図17がCX10047の構成です。いずれもミニフラット・パッケージです。

### ● サンプル&ホールド回路

CCDからの出力信号は、写真4、写真6に示すように、受光素子からの出力間にPGパルスが重畳した波形になっています。したがって、S&Hパルス SHD、SHPを使用してビデオ信号を取り出します。このタイミングを図18に、そのS&H出力波形を写真5に示

〈表3〉 NTSC方式の規格

走査線数/フレーム	525本
垂直(フィールド)周波数 $f_v$	59.94Hz
水平(ライン)周波数 $f_H$	15.734264kHz
水平(ライン)周期	63.5556μs
水平ブランキング H. BL	10.9±0.2μs
フロント・ポーチ	1.27μs~2.22μs
水平同期 H. Sync.	4.7±0.1μs
垂直ブランキング V. BL	19~21H
等価パルス Eq. pulse	3H 2.3±0.1μs
垂直同期 V. Sync.	3H 27.1μs
カラー副搬送波 $f_{sc}$	3.579545MHz, $f_{sc} = \frac{455}{2} f_H$
試作機の基準発振 $f_{ref}$	28.63636MHz = 1,820 $f_H$
CX7930のクロック入力 CK	14.31818MHz = 910 $f_H = \frac{f_{ref}}{2}$

します。

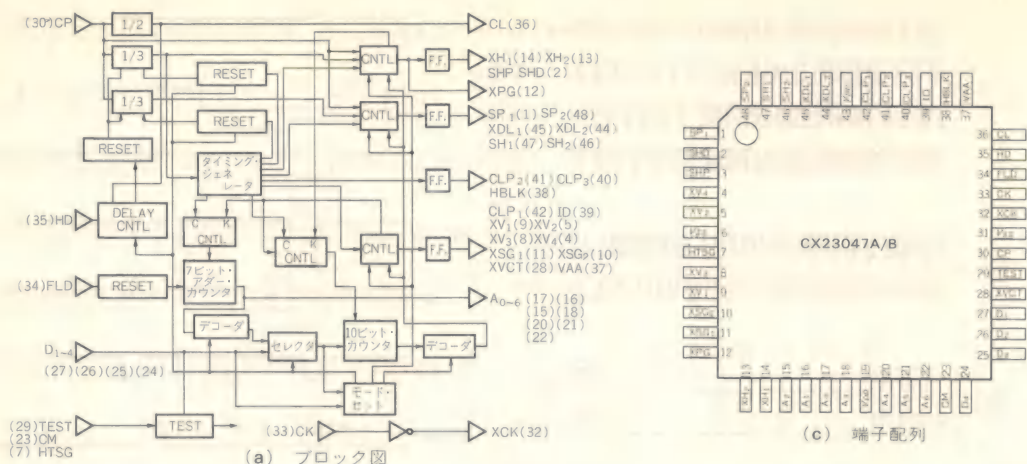
なお、図15におけるL<sub>2</sub>はローパス・フィルタを構成しており、S&Hパルスによって発生するクロック・ノイズを除去しています。

またVR<sub>6</sub>は、差動アンプのバイアスを決めるボリュームです。チェック端子TP<sub>1</sub>にて、DC電圧が5.2Vになるよう調整します。


### ● ビデオ信号処理

IC<sub>3</sub>のCX10047は、S&Hされたビデオ信号を各種処理することにより、コンポジット・ビデオ信号を得るためのプロセス回路です。図15において、クランプ・パルス CLPを使用して、クランプ動作(直流再生)を行い、ビデオ信号に重畳しているハム・ノイズなどを除去し、黒レベルの再生を行います。





〈図13〉 CX23047の構成

端子番号	記号名	I/O	端 子 説 明
4, 5, 8, 9	XV <sub>1</sub> XV <sub>4</sub>	O	イメージ駆動パルス, 反転型ドライバを付加してCCDイメージ(IX018,021)を駆動。
10, 11	XSG <sub>1,2</sub>	O	
12	XPG	O	
13, 14	XH <sub>1,2</sub>	O	
15~18 20~22	A <sub>0</sub> ~A <sub>6</sub>	O	外部ROM用アドレス出力, A <sub>6</sub> がMSB
23	CM	I	テスト端子, 通常GND
24~27	D <sub>4</sub> ~D <sub>1</sub>	I	外部ROM入力端子, 外部ROMを使用しない場合は, プルアップあるいはプルダウンによってモード設定が可能. ROMを使用しない場合: D <sub>1</sub> 常にGND D <sub>2</sub> -GND: 白黒モード, V <sub>cc</sub> : カラーモード D <sub>3</sub> 常にV <sub>cc</sub> D <sub>4</sub> -GND: CCIR, V <sub>cc</sub> : NTSC
28	XVCT	O	ROM(MB7052)電源スイッチング・パルス. スイッチングを行う場合はPNPトランジスタを付加しなければならない。
29	TEST	I	テスト端子, 通常GND
30	CP	I	クロック入力, NTSC: 28.6364MHz, CCIR: 28.3750MHz
32	XCK	O	インバータ回路 CK  XCK
33	CK	I	
34, 35	FLD, HD	I	同期信号入力, CL(36ピン)の立ち下がりを取り込まれる。
36	CL	O	SYNCジェネレータ(CX7930)用クロック出力, CPの半分の周波数。
37	VAA	O	CCDイメージ出力の垂直有効領域, クランプ回路においてCLP <sub>1</sub> とともに用いる。
38	HBLK	O	CCDイメージ出力の水平有効領域, フリ・プランキングに使用する。
39	ID	O	R/Bライン識別信号, Bライン "H", Rライン "L"
40, 41	CLP <sub>2,3</sub>	O	グランプ・パルス, 連続パルス。
42	CLP <sub>1</sub>	O	グランプ・パルス, CCD出力オフティカル・ブラック部クランプ・パルス。
44, 45	XDL <sub>1,2</sub>	O	1Hディレイ・ライン(CX23039)駆動パルス。
46, 47	SH <sub>1,2</sub>	O	1Hディレイ・ライン(CX23039)用サンプル&ホールド・パルス。
48, 1	SP <sub>1,2</sub>	O	色分離用サンプル&ホールド・パルス, 欠陥補正機能をもつ。
2	SHD	O	イメージ出力用サンプル&ホールド・パルス, 欠陥補正機能をもつ。
3	SHP	O	イメージ出力のプリチャージ・レベルをサンプル&ホールドするパルス。
7	HTSG	I	テスト端子, 通常GND
6, 31	V <sub>SS</sub>	I	接地端子。
19, 43	V <sub>DD</sub>	I	+5V電源端子。

(b) 端子機能



また、このほかの調整箇所として以下のようなものがあります。

(a) CXB0026M  
(CCD 水平ドライバ)

(b) SN75361A  
(TTL-SM ドライバ)

ビデオ信号のホワイト・レベルをWHT CLIP  $VR_2$ にて規定することにより、コンポジット・ビデオ信号のダイナミック・レンジを決めます。一般には1.1  $V_{P-P}$ 程度にします。

ビデオのブランキング期間(写真6参照)には、いろいろなクロック・ノイズが含まれています。したがって、ブランキング・パルス BLKを使用して、波形の整形を行います。

PEDESTAL LEVEL  $VR_8$ を調整することにより、ビデオ信号の黒レベルを決めます。このレベルは、CCDをしゃ光状態にしたとき、**約20mV<sub>P-P</sub>**になるように調整してください〔写真7(a)参照〕。

ブランキング期間に同期信号SYNCをミックスする

PRE VIDEO

L<sub>3</sub> 3.3μH

L<sub>4</sub> 3.3μH

220Ω

1μF 50V

PULSE GND

SHP

IC<sub>1</sub> CX10045

ノイズ・パルス除去回路

V<sub>CC</sub>

16

13

9

10μF 16V

NC

11

10

6

3

7

5

8

12

20pF

2.2kΩ

2SK192AGR

5

4

15

1

2

5

2.2kΩ

20pF

0.01μF

0.01μF

0.1μF (SC)

Tr<sub>2</sub>

Tr<sub>1</sub>

VR<sub>6</sub>

5kΩ

IC<sub>2</sub> CX7390

+9.5V

D<sub>4</sub>

+5V

28

V<sub>DD</sub>

BLKG

OUT

4

26

O/E

OUT

5

4

20

HD

OUT

EXT

IN

1

10

100k

CK

IN

4

VR

IN

23

100k

HRIN

LALTR

15

16

MODE2

TEST

MODE1

GND

14

TP<sub>1</sub>

IC<sub>3</sub> CX10047

0.2Vp.p

0.1μF (SC)

82

10k

Tr<sub>4</sub> 2SC1815Y

C<sub>9</sub> 47μF 6.3V

2.2kΩ

10k

3.3kΩ

10pF

82pF

1kΩ

L<sub>2</sub> 33μH

1μF 50V

3.3kΩ

220pF

CLP

1

2

3

1kΩ

15kΩ

5kΩ

3.3kΩ

3.9kΩ

2.2kΩ

VR<sub>1</sub>

VR<sub>2</sub>

VR<sub>5</sub>

VR<sub>3</sub>

10kΩ

100pF

10kΩ

15pF

3.3kΩ

3.3kΩ

100μF 16V

Tr<sub>5</sub> 2SC1815Y

Tr<sub>6</sub> 2SC1509

75Ω

1.5kΩ

470Ω

VR<sub>3</sub>: SYNC LEVEL

ビデオ出力

CHASSIS GND

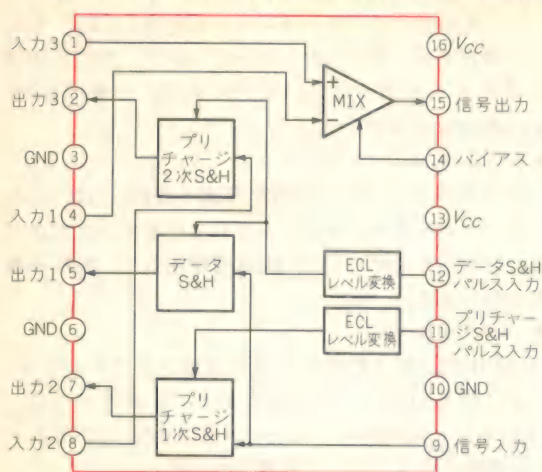
VR<sub>2</sub>: WHT CLIP

VR<sub>5</sub>: PEDESTAL LEVEL

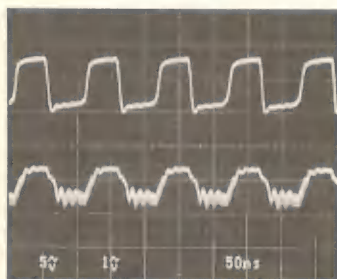
VR<sub>1</sub>: γ CONTROL



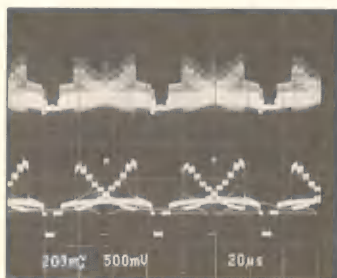
〈図16〉 CCDカメラ信号サンプル&ホールド用IC CX10045の構成



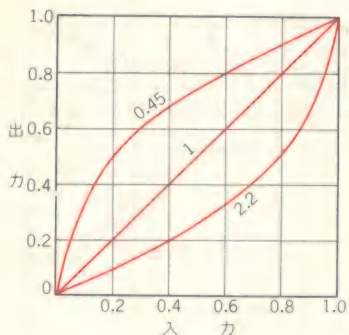
〈写真5〉  
PGとCCD出力信号



〈写真6〉  
S&H出力とビデオ出力

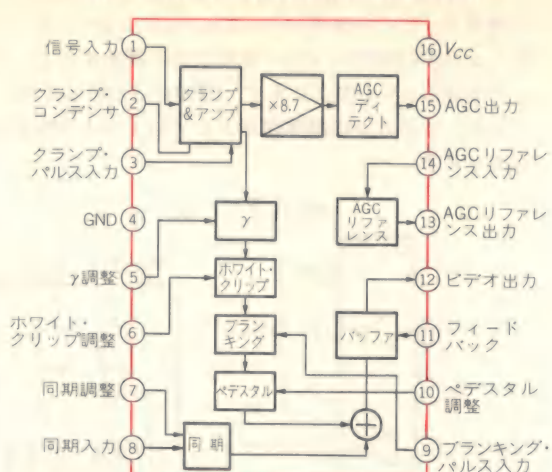


〈図19〉  $\gamma$ カーブ

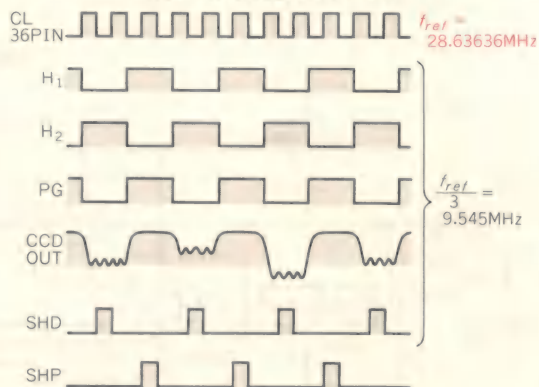


テレビのブラウン管は  $\gamma=2.2$  であるため、放送用のTVカメラは  $\gamma=0.45$  で設計されている。

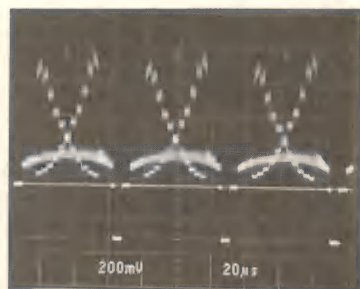
〈図17〉 B/Wカメラ信号処理用IC CX10047の構成



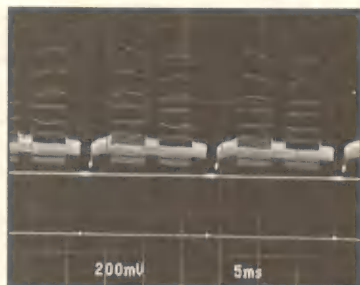
〈図18〉 水平転送のタイミング



〈写真7〉 ビデオ出力



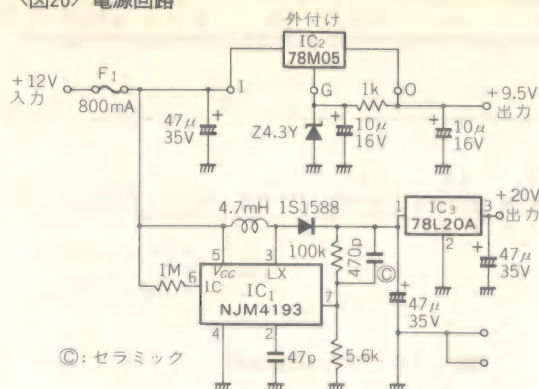
(a) 水平期間：H



(b) 垂直期間：V



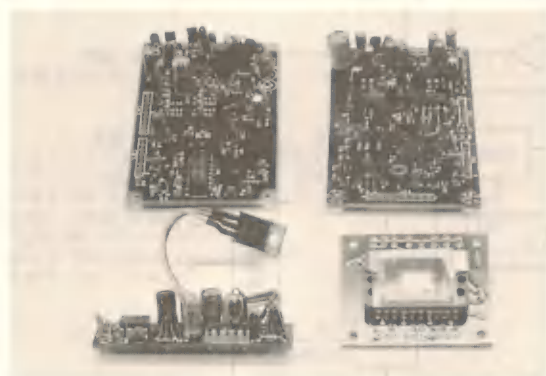
〈図20〉 電源回路



©: セラミック



〈写真9〉 製作したカメラの外観



〈写真8〉 各回路基板



〈写真10〉 モニタの画面

ことにより、コンポジット・ビデオ信号を作ります。

ここではSYNC LEVEL  $VR_3$  にて $300mV_{P-P}$ に調整します〔写真7(a)参照〕。

#### ▶ 出力回路

$Tr_5$ と $Tr_6$ は、コンポジット・ビデオ信号を $75\Omega$ の出力インピーダンスに変換する回路です。したがって、 $VR_1 \sim VR_3$ の調整を行うときは、必ずビデオ出力を $75\Omega$ で終端します。

基本的には以上の調整でカメラは完成しますが、このシステムでは+12V単一電源で動作するようにするために、DC-DCコンバータを内蔵させました。この構成を図20に示します。内訳は、+9.5Vと+20VのDC-DCコンバータです。

なお、 $IC_2$ の78M05は発熱(約1W)しますので、放熱が必要です。

今回製作した各基板を写真8に、カメラの外観を写真9、モニタの画面を写真10に示します。

### 外付けオプションの製作

#### ● ストロボとのインターフェース

CCDカメラには、受光部の信号電荷を1/60秒ごと

に読み出すフィールド蓄積モードと、1/30秒ごとに読み出すフレーム蓄積モードがあります。

今回試作したCCDカメラは、**フレーム蓄積モード**です。

言い換えれば、**1/30秒のシャッター・カメラ**として動作していますので、モータで回転しているものや、ベルト・コンベアで高速移動している物体などの速い動きを捕えることはできません。

これらの被写体を静止画像として取り込むためには、ストロボとビデオ・メモリとが必要になります。

#### ▶ ストロボの発光周波数

- (1) 内部同期方式：1～500Hzの繰り返し周波数で点灯するもので、**マルチストロボ**といえます。
- (2) 外部同期方式：主にテレビ・カメラのVDまたはV SYNCと同期をとって1/60秒周期で発光します。

VDからの発光タイミングを可変できるものもあります。

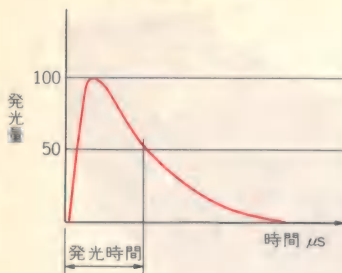
#### ▶ 発光時間

$1\mu s \sim 100\mu s$ (1/1,000,000秒～1/10,000秒)があります。この時間は半値幅といって、図21のように最大発光量の1/2に落ちたところの時間で示されています。

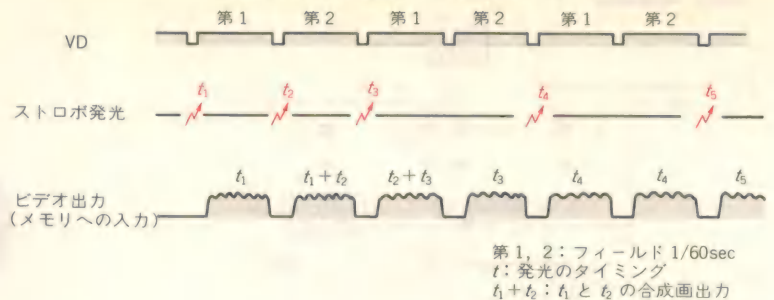
#### ▶ 発光光量



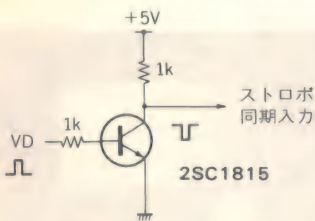
〈図21〉 発光特性



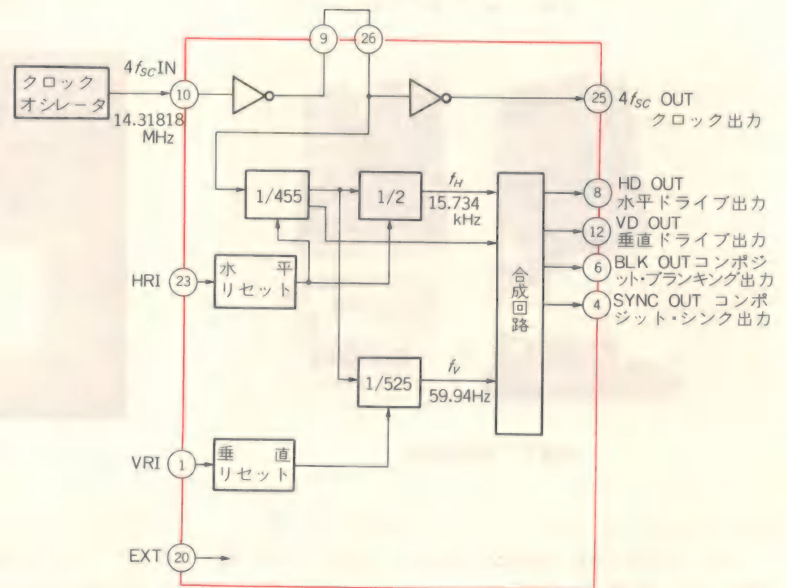
〈図22〉 ビデオ出力(フレーム蓄積モード)



〈図23〉 ストロボとの  
インターフェース回路



〈図24〉 CX7930Aの内部構成図



ビデオ用のストロボは、発光光量のばらつきが±10%以内に収まっています。光量は非常に大きい発光が得られますので、レンズの絞りで適正值になるようにモニターを見ながら、光量を制限する必要があります。

また光源が赤外光を含んでいると、スミアの原因になりますので、レンズ前面に赤外カット・フィルタを取り付けると画質がよくなります。

### ● メモリへの取り込み

#### ▶ 取り込みタイミング

ストロボによって得られたビデオ信号を、メモリに取り込むタイミングはとくに大切です。図22に示すように、フレーム蓄積モードのCCDカメラに、1/60秒ごとでストロボ発光を行うと、ビデオ出力には2画面分の合成画がメモリに取り込まれます。

1画面だけをメモリに取り込みたいときは、ストロボの点灯を1回だけにするか、1/30秒ごとにストロボ発光を行います。図23にストロボとのインターフェース例を示します。市販のストロボ装置(ストロボ・スコープ)には外部同期端子が出ていますので、この入力

へ信号を送り込みます。

#### ▶ サンプリング・クロック

CCDカメラの水平分解能は受光素子の数で決まりますので、必要以上に高い周波数にする必要はありません。今回使用しているICX018CLは、水平方向510個で、水平転送クロック $H_{01}$ 、 $H_{02}$ は、

$$(f_{ref}/3) = (28.63636\text{MHz}/3) \\ = 9.545\text{MHz}$$

です。

このクロックはIC<sub>0</sub> CX23047Bから取り出せますので、受光素子に対応したメモリ取り込みができます。

### ● 同期信号発生器

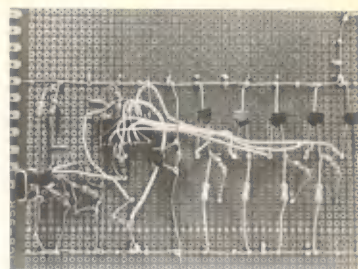
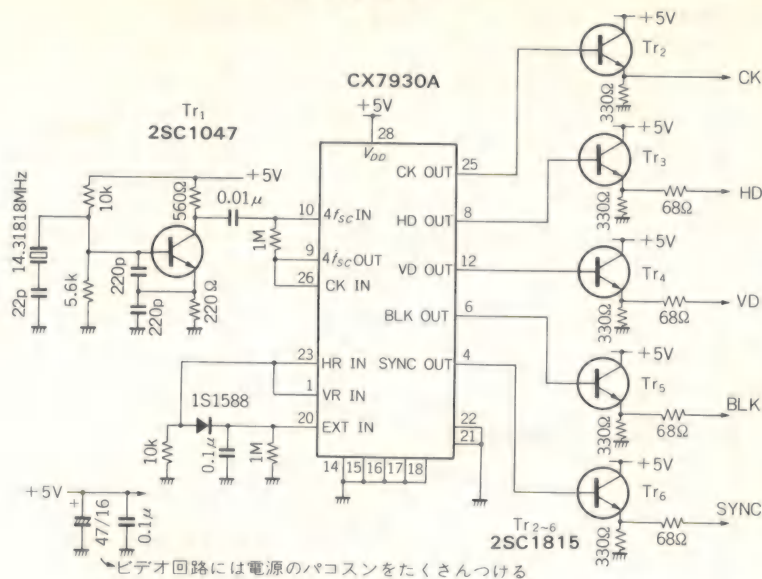
テレビ・カメラに外部同期をかけたり、画像メモリにビデオ信号を取り込んだあとの信号源として有効な、同期信号発生器を1台作っておくととにかく便利です。

そこで、CCDカメラの製作にも使用したCX7930Aを使って設計してみました。図24がそのブロック図です。

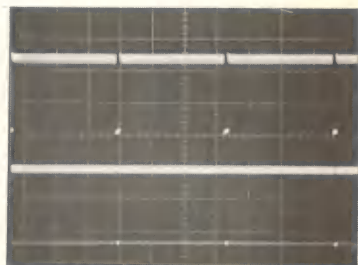
水平周波数 $f_H$ は、



〈図25〉 同期信号発生器



〈写真11〉 同期信号発生器の実験基板



〈写真12〉 VDとSYNCの出力波形

$$f_H = \frac{14.31818\text{MHz}}{455 \times 2} = 15.734\text{kHz}$$

垂直周波数 $f_V$ は、

$$f_V = \frac{14.31818\text{MHz}}{455 \times 525} = 59.94\text{Hz}$$

となります。

図25が実際の回路です。

Tr<sub>1</sub>の基準発振器は、無調整で済むようにクリスタル発振器にしました。なおCX7930Aは、ミニフラット・パッケージですので、はんだ付けには十分注意してください。

信号出力には、バッファ・トランジスタを入れて75Ω系の負荷にも対応できるようにしました。

テレビ・カメラに外部同期をかけるときは、HD、VD、またはSYNCを使用すると簡単に同期がかかります。

写真11が製作した外観で、写真12がVDとSYNCの出力波形です。

## テレビ・カメラの光学系

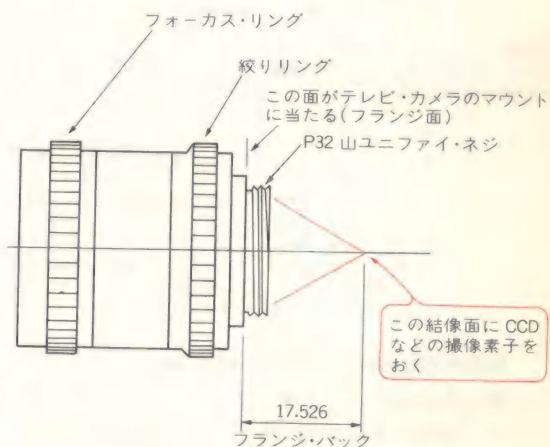
画像処理に使用する光学系は、被写体の大きさ、カメラからの距離、色(波長)、明るさ、解像度、カメラのイメージ・サイズなどによっていろいろと組み合わせが考えられます。

まず、市販されている工業用テレビ・カメラのレンズ・マウントは、Cマウントが多く使用されており、図26のようになっています。

### ● 固定焦点レンズ

一般的に最も使用されているレンズは、表4に示す

〈図26〉 固定焦点レンズの形状とCマウント規格



ように焦点距離によっていろいろなものがあります。

表4において、 $f$  12.5~16mmを標準レンズと称しています。したがって、それ以下の焦点距離のレンズは、広角(ワイド)レンズ、それ以上のレンズは望遠(テレ)レンズと称しています。

画角は、図27に示すように、レンズの焦点距離と撮像素子の大きさとの比から計算されます。

口径比( $F$ ナンバ)とは、レンズの明るさを表し、 $F$ ナンバと明るさは反比例します。 $F$ ナンバは、絞りリングで可変することができ、1ステップ絞ると光量は1/2になります。

〔例〕

$F$ ナンバ 1.4 → 2 → 2.8

(1ステップは $\sqrt{2}$ 倍)



〈表4〉  
Cマウント・  
レンズ(RIOMA)：  
扱い、プロテック・  
ジャパン  
☎045(574)3876

2/3インチ用

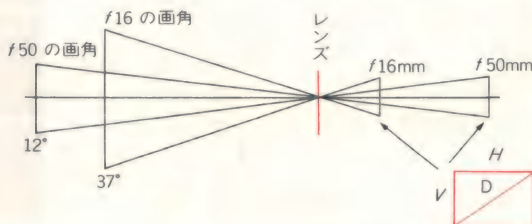
型式	焦点距離 $f$ (mm)	口径比 $F$ ナンバ	画角	最短撮影距離 (m)
SX414R	4	1.4	114°	固定
SX813R	8	1.3	72°	固定
SX1616R	16	1.4	37°	0.5
LX2518R	25	1.4	24°	0.5
LX5013R	50	1.3	12°	1.0
S6Z12575	12.5~75	1.8	W: 48°, T: 8°	1.0
S10Z11110	11~110	1.8	W: 56°, T: 6°	1.3

固定焦点レンズ

ズーム・レンズ

W:ワイド, T:テレ

〈図27〉 焦点距離 $f$ と画角

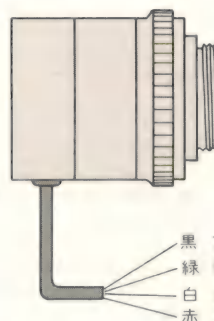


画角はイメージ・サイズの対角線に対する  
角度で表す。

イメージ・サイズ  
2/3インチ撮像素子の場合  
H ヨコ 8.8mm  
V タテ 6.6mm  
D 対角線 11mm

〈図28〉  
オート・アイリス・レンズの仕様  
と接続方法(RIOMA)

DC 入力	DC8~16V
入力信号	1.0V <sub>p-p</sub> コンボジット
入力インピーダンス	高インピーダンス
動作方式	平均ピーク測光
動作温度範囲	-20°C~+60°C
応答スピード	1.5 秒以内
動作精度	1.0V <sub>p-p</sub> ±10%



テレビ・カメラ  
に接続

像面照度 400 lx→200 lx→100 lx

最短撮影距離とは、フォーカス・リングを回してフォーカス調整をするとき、フォーカス調整が可能なレンズから被写体までの最短距離をいいます。

#### ● ズーム・レンズ

表4に示したように、焦点距離を変化させることにより、結像倍率を変えるレンズをズーム・レンズといいます。 $f$  12.5~75mmのレンズをその比をとって6倍ズーム・レンズといいます。

$$75 \div 12.5 = 6$$

被写体の距離や大きさが変化するときにはズーム・レンズが便利です。

#### ● オート・アイリス・レンズ

通常のレンズは、絞りを手で回して光量の調整をしますので、マニュアル・アイリス・レンズといいます。

これに対し、図28に示すように、テレビ・カメラから電源とビデオ信号を供給してもらい、ビデオ信号が規定値(通常は1.0V<sub>p-p</sub>)より大きい時は絞りが閉じる

方向に動き、規定値より小さい時は絞りが開く方向に自動的に制御して、被写体の明るさが変化しても、常に一定のビデオ出力が得られるレンズをオート・アイリス・レンズといいます。

オート・アイリス・レンズは、モータ、制御回路を内蔵しています。被写体の明るさが変化するときには、オート・アイリス・レンズが便利です。

#### ● モータ・ドライブ・レンズ

ズーム・レンズにおいて、フォーカス・リング、ズーム・リング、アイリス・リングのそれぞれにモータが付けられているレンズをモータ・ドライブ・レンズといいます。

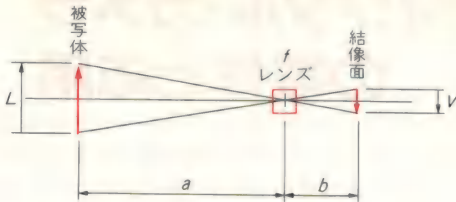
このレンズは、離れた場所からリモート・コントロールができますので、テレビ・カメラが離れた場所にあるときに使用します。

#### ● Cマウント・アダプタ

35mm一眼レフ・カメラ用の各種レンズを、Cマウント・レンズに変換するためのアダプタがCマウント・ア



〈図29〉 結像倍率



・結像の公式  $\frac{1}{a} + \frac{1}{b} = \frac{1}{f}$

・結像倍率  $m = \frac{b}{a} = \frac{V}{L}$

通常は  $a \gg b$  の関係にあるので  $b = f$  となり,  $m = \frac{b}{a} = \frac{f}{a}$  となる

アダプタです。

だれでも変換レンズの1本ぐらいはお持ちでしょうから、このアダプタを購入すると便利です。アダプタは交換レンズ・メーカ、アクセサリ・メーカから入手できます。

### ● 被写体照度と像面照度

被写体の明るさがわかれば、使用するレンズとテレビ・カメラを決めることができます。

像面照度  $E_c$  は、

$$E_c = \frac{T \times R}{4 \times F^2 \times (m + 1)^2} \times E_s$$

$T$ : レンズの透過率

$R$ : 被写体の反射率

$F$ :  $F$  ナンバ

$m$ : 結像倍率

$E_s$ : 被写体照度

[例]

$T=80\%$ ,  $R=50\%$ ,  $F=1.4$ ,  $m=1$ ,  $E_s=1000$  lx としたとき、

$$E_c = \frac{0.8 \times 0.5}{4 \times 1.4^2 \times (1 + 1)^2} \times 1000 = 13 \text{ lx}$$

となります。

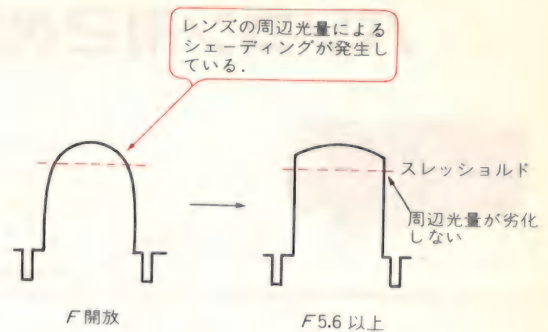
一般の CCD カメラは 5 lx 程度まで十分な規定出力が得られますので、この場合は絞りを 1～2 ステップ閉じて使用します。

### ● 実際の使用例

通常の場合、レンズは遠くのことを撮るために設計されていますので、画像処理のように近くにある被写体を撮る場合は少し工夫が必要です。

例えば、高さ 5 cm の大きさのものを、 $f25\text{mm}$  のレ

〈図30〉  $F$  ナンバとビデオ出力



ンズにて画面いっぱいに撮るには、図27と図29の式より、

$$\text{結像倍率 } m = \frac{b}{a} = \frac{V}{L} = \frac{6.6}{50} = 0.132$$

結像の公式  $(1/a) + (1/b) = 1/f$  より、

$$b = \frac{af}{(a-f)} = \frac{(b/0.132) \times 25}{(b/0.132) - 25} = 28.3 \text{ mm}$$

$$28.3 - 25 = 3.3 \text{ mm}$$

となり、レンズと撮像素子の距離を無限遠物体を撮る場合と比べて 3.3mm 離す必要があります。

### ● 2 値化するときの注意点

ビデオ信号を処理するとき、スレッシュホールドを決めて固定 2 値化することがよく行われます。

このとき、レンズ絞りを開放付近で使用すると、画面中心に比べて画面周辺の光量が低下するため、ビデオ信号にシェーディングが発生して、周辺部の 2 値化がうまくいきません。

この場合は照明をもっと明るくして、 $F5.6$  以上にすると図30のようにきれいな出力が得られます。

以上いろいろと書きましたが、画像処理を行ううえで、照明、レンズ、TVカメラの選び方によって性能が決まる部分がたくさんありますので、その選択には十分注意してください。

### ● 参考・引用文献

- (1) 谷川 紘; 他; CCD型素子, テレビジョン学会誌, 1986年11月号.
- (2) 竹本一八男; MOS型素子, テレビジョン学会誌, 1986年11月号.
- (3)\*SONY, CCD撮像素子DATA BOOK, 1987年版.



# パソコン用2次元デジタル・カメラ の製作と応用



寺本三省

CCDなどの代わりに、オブティックRAMを使っても2次元のカメラを製作できます。これは回路も簡単で、パソコンにもフィットします。カメラの製作例と画像認識への応用を紹介します。

固体撮像素子といえばCCD，MOSタイプが代表的で、これ以外の素子はあまり知られていません。

ここで紹介するパソコン直結の2次元カメラは、前述の正統的な撮像素子とは、内容と発想が大きく異なる撮像素子を用いて製作しました。この素子は、米国マイクロン・テクノロジー社〔扱い：トマスエレクトロン(株)〕によって開発され、同社によって“オブティックRAM”の商品名で製造されています。デバイスの品番はIS32となっています。

この素子を採用したカメラは、米国ではIBMやアップルのパソコン用にマイクロン・テクノロジー社が、日本では各社パソコン用に(株)デジック(☎0899-24-0914)が開発し、製造販売を行っています。

オブティックRAMのRAMは、ランダム・アクセス・メモリの略であり、光に関連したメモリであることが連想されますが、まったくそのとおりで、完全なメモリなのです。

## 撮像素子オブティックRAM

IS32の外観は写真1のように、セラミック・パッケージの上部が石英ガラスでシールされており、内部のチップが見える構造です。ピン数が16であることから、小型のEP-ROMを感じさせます。

IS32の撮像素子として主要な仕様を表1に示します。  
● IS32の動作原理

さて、この素子はどのようなしくみで撮像素子として動作するのか、また光とメモリの関係がどこにあるのでしょうか。これらを知ることは、カメラを製作す

るためだけでなく、新しい機能を発想するうえでも重要です。

図1はよく見かけるダイナミックRAMのセル等価図ですが、ビット線のデータがワード線をアクティブにするとQが導通し、 $C_s$ へデータが蓄えられます。 $C_s$ はpn接合の接合容量を形成しており、 $C_s$ へは電荷が蓄積されることになります。

読み出し時は、逆に $C_s$ の電荷に対応したデータがビット線に現れることになります。この場合、書き込みと読み出しの時間間隔が短いときはこのままでメモリとして成り立ちますが、長い時間差となると、 $C_s$ の電荷は $C_s$ やQのリークが作用して消滅し、読み出しでも偽のデータとなり、メモリ機能を失うことになります。

このために、DRAMの内部には $C_s$ の内容を呼び出し、残留電荷量を判定しながら新しく再チャージを行う機能をもたせています。これをリフレッシュと呼んでおり、定期的に外部から制御して行います。

さて、ここでもう一度 $C_s$ のリークについて考えます。 $C_s$ のpn接合へ光を照射するとどうでしょう。これについては、フォト・ダイオードを想起していただくとも明瞭になりますが、電子が励起されエネルギーが発生し、これが光電流となりますから、 $C_s$ のリークを増大させるように働きます。

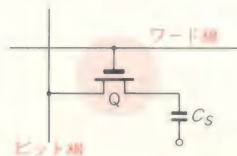
勘の良い読者の方は、説明の意図を気付かれたと思いますが、もう少しすすめてみましょう。

メモリ・セルへ論理“1”を書き( $C_s$ へ電荷をチャージ)、電荷の残留限度の時間内にこれを読み出せば、元の“1”が得られます。しかし、 $C_s$ へ光が照射さ

〈表1〉 IS32の仕様

セル・アレイ	128行×256列の電氣的アクセス可能素子群を左右に配置、サイズは4420×876.8 $\mu\text{m}^2$ 物理的配列構成は514列×129行の格子状
画素サイズ	6.4×6.4 $\mu\text{m}^2$
水平方向の画素ピッチ	8.6 $\mu\text{m}$
垂直方向の画素ピッチ	6.8 $\mu\text{m}$
左右セル・アレイ間のスペース	120 $\mu\text{m}$
ガラス表面とチップの間隔	900 $\mu\text{m}\pm 20\mu\text{m}$

〈図1〉  
メモリ・セルの等価回路



〈写真1〉 IS32の外観  
(マイクロン・テクノロジー社)



れているとすると、電荷の放電が助長されますから、先と同一時間に読み出しても“1”ではなく、“0”を得るかもしれません。

このような現象をソフト・エラーと呼び、通常のメモリでは有害となりますが、見方を変えれば、このことはセルが光の存在を検知し、それを記憶したとも考えることができます。

メモリ内部には、このような性質をもつセルが2次元的に多数配列してありますから、これら全体に“1”を書き込んで光のパターンを照射し、個々のセル内容を読み出せば、光の強い所は“0”，弱い所は“1”というようになり、光パターンに相似した分布データを得ることができでしょう。

光によるソフト・エラーの発生原理を逆手にとって、撮像素子として開発されたのがオプティックRAM IS32なのです。

とはいっても、IS32の内部機構と基本的動作特性は、図2に見られるように、標準の64KDRAMと大差なく、その限りでは何の変哲ありません。違っているのは、写真1のようにパッケージの天井を光に対し吹き抜け構造としていること、それと1番ピンの機能です。

### ● IS32の撮影シーケンス

IS32がDRAM構造であることで、よりコンピュータに接近したカメラの製作が可能です。すなわち、IS32は完全なデジタル素子であり、コンピュータで

最も多く使われているメモリそのものだからです。

ここで先の原理をふまえ、IS32の撮影シーケンスを追ってみましょう。

はじめに**全セルに“1”を書き込みます。これが撮像準備**となります。内容を失わないために、リフレッシュを繰り返します。

撮影は一般の写真と同様に、被写体からの反射光をレンズで集光し、感光媒体へ投射して行います。この場合、セル・アレイが感光媒体となります。

撮影の開始はリフレッシュの停止で行います。セルに蓄えられた電荷は、光の強い部分から減少していきます。

適当な時間後リフレッシュを再開します。素子内部ではセルごとに残った電荷を、あらかじめ定められたスレッシュホールド・レベルと比較し、レベルより高いと“1”，低いと“0”として新しく書き換えていきます。つまり、この動作は**1ビットのA-D変換**と同様で、この撮像素子はA-Dコンバータ内蔵型といえます。

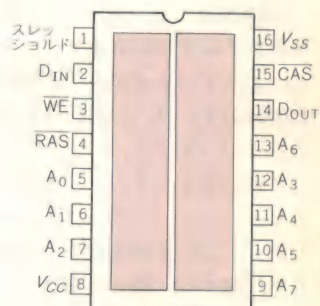
結果は投射された像の明暗に対応して、明るい部分のセルは“0”が多く、暗い部分は“1”が多く分布した形で記憶され、像が定着したことになります。

これを読み出せば、コンピュータに適合する、デジタル化された画像データが得られます。

ここでの書き込み、読み出し、リフレッシュは、通常のDRAMで行うのと同じに行えばよいのです。図3を参考にしてください。

〈図2〉 IS32の構成

- ・デュアル128×256エレメント・アレイ
- ・32,768エレメント/アレイ
- ・光感度調整可能
- ・可変スキャン・レート
- ・可変スレッシュホールド電圧
- ・出力ラッチ付き
- ・5V±5%電源
- ・TTLコンパチブル

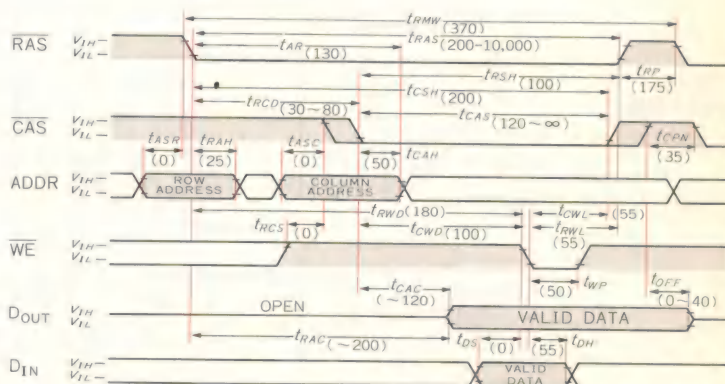


(a) 特徴

(b) ピン接続

〈図3〉

IS32のタイムチャート(リード・ライト・サイクル, リード・モディファイ・ライト・サイクル)





## ● IS32のセル配置

表1からわかるように、チップ上に128行×256列のセル・アレイが2群配置され、アレイとアレイの間は120 $\mu\text{m}$ のスペーシングで仕切られています。

このスペーシングが撮影ではデッド・ゾーンになりますから、65,536個のセルの半分以上が使用できなくなります。このデッド・ゾーンをキャンセルして、アレイ全部を有効に活用できるカメラが、(株)デジックから出ていますが、精密な機構が必要で自作は大変ですから、今回は半分で我慢することにします。

図4にセルの物理的配置を示します。グリッド内に $R_n$ 、 $C_n$ と記された部分がセルの存在する部分で、後はスペース領域となります。

セルはかなり間引きされた状態で分布しています。これはこの素子がメモリ構造であることからで、先のデッド・ゾーンについても同様のことがいえます。

さて、これをそのまま使って忠実に再現すると、歯抜けした画像になって支障をきたします。これを補う方法として補間という技法を使います。

これはこの素子に限らず、アナログ表現を行うディジタル機器ではさけて通れないことです。ディジタル方式の高速サンプリング・スコープのように補間を行わないと、線形の波形も点々の表示となってしまいます。

セルの物理配置がわかると、次は素子の論理アドレスがこれとどのような関係になっているかです。

DRAMはアドレス・ピンの節約を行うために、行と列のアドレスを定めます。これを時分割で指定するようになっていますが、この場合は行0列0の位置が物理空間の $Y_0$ 、 $X_0$ となるとよいのですが、そのようにはなっていません。

この対策として、メーカーのマイクロン社では図5のアドレス・デスクランブルなる回路を公表しています。しかし、これを使えばすべて解決かというそうではなく、図4は、図5で補正したアドレスで記したもので、まだ補正が必要であることがわかります。

今回のカメラでは、先の補間とアドレス補正を後述

のソフトウェアで行っていますから参考にしてください。

## ● IS32の光学特性

図6に主要な光学特性を示します。まず図(a)は分光感度曲線です。この素子は長波長域に感度が伸びており、700nm～800nmに最高感度点があり、短波長に向かって低くなっていることがわかります。

色彩では赤系に高く、青系に低いといえますが、人間の視感度領域はカバーしていますから、普通の撮影には支障ありません。

図(b)はIS32の1番ピンのスレッシュホールド電圧値に対する、セルの感度変化を示しています。この電圧は1.5V～3Vの範囲に限定して指定されており、これ以外では正常な動作は期待できません。

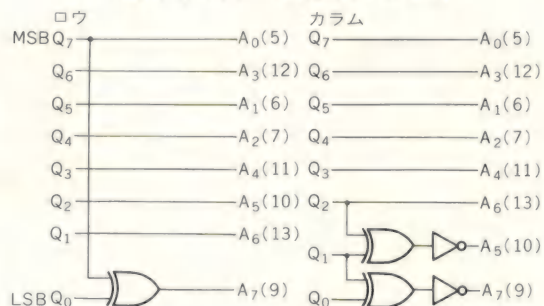
この図から、電圧の変化範囲での感度の比率は2以下で、大きな変化は期待できません。実際には、露光の度合いで調整するのがベターといえます。

次に重要なのが、セル感度のばらつきです。この傾向が図(c)です。この図は露光タイムの変化に対する、セル感度の推移を相対値で示しています。実線が平均値で、点がばらつきの分布状態を表しています。

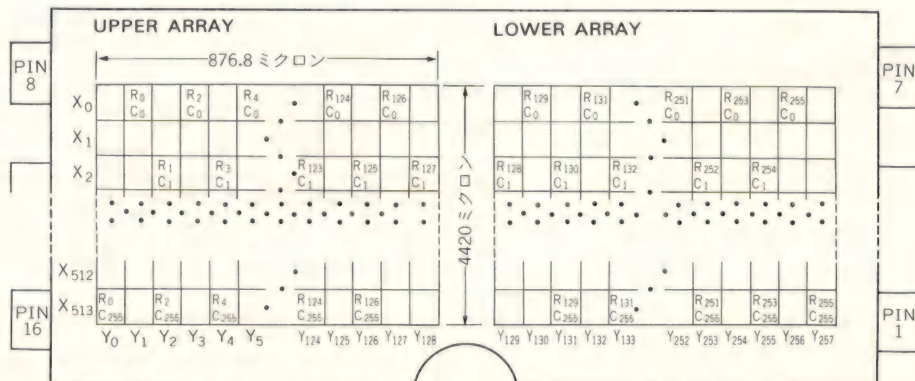
図より約20%の不均一性があることがわかります。これはCCDなどの素子に比べ劣りますが、被写体のコントラストを大きくすることでカバーできます。

図(d)は素子を暗におき、データを書いた後のデータ保持時間を、温度の変化に対しプロットしています。

〈図5〉アドレス・デスクランブル回路

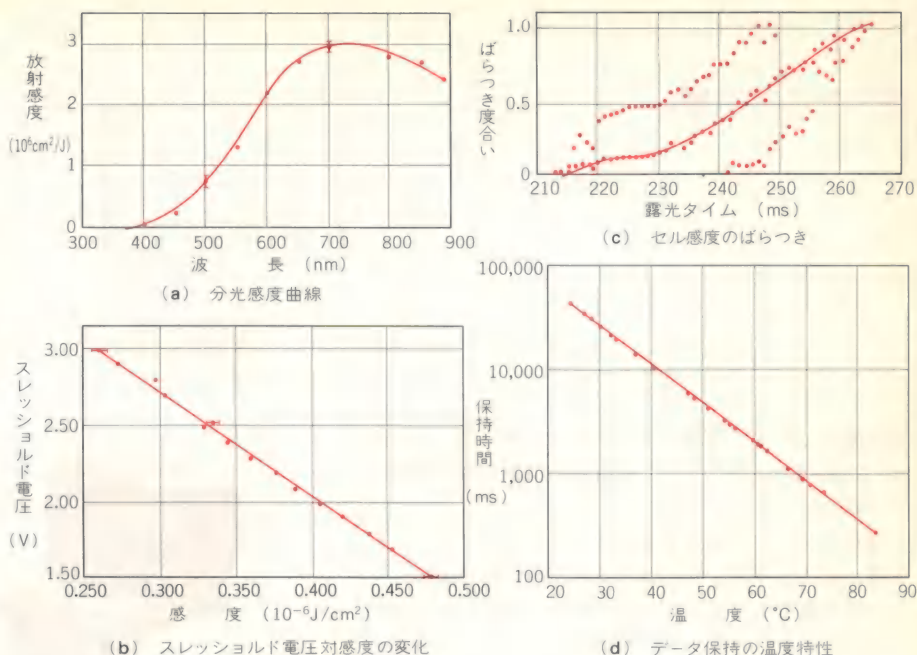


〈図4〉セルの配置



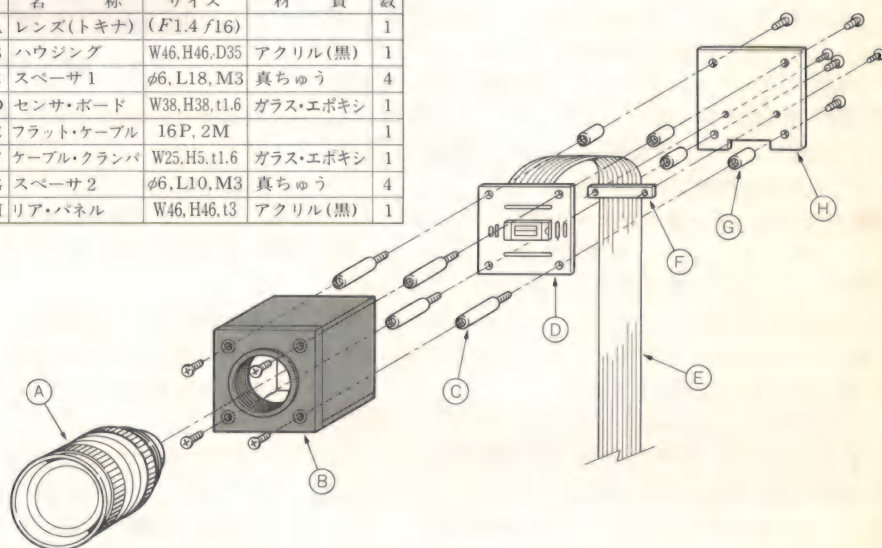


〈図6〉  
IS32の光学特性



〈図7〉  
カメラ部の組み立て図

	名 称	サイズ	材 質	数
A	レンズ(トキナ)	(F1.4 f16)		1
B	ハウジング	W46,H46,D35	アクリル(黒)	1
C	スペーサ1	φ6, L18, M3	真ちゅう	4
D	センサ・ボード	W38,H38,t1.6	ガラス・エポキシ	1
E	フラット・ケーブル	16P, 2M		1
F	ケーブル・クランプ	W25,H5,t1.6	ガラス・エポキシ	1
G	スペーサ2	φ6, L10, M3	真ちゅう	4
H	リア・パネル	W46,H46,t3	アクリル(黒)	1



温度依存性がほかの半導体と同じ傾向であることがわかります。

この図は、露光時間の上限を設定するうえで重要です。

## デジタル・カメラの製作法

今回はできるだけ簡単なくみで製作するということと、コンピュータ直結をテーマにしたために、極力ハード部分を少なくし、CPUやソフトに依存することにした。

接続対象のコンピュータは、日電のパソコン

PC9801としています。

### ● 撮像部のしくみ

撮像部はIS32とレンズの収容だけにして、小型化を図りました。この部分が図7です。シンプルですから容易に自作できると思います。

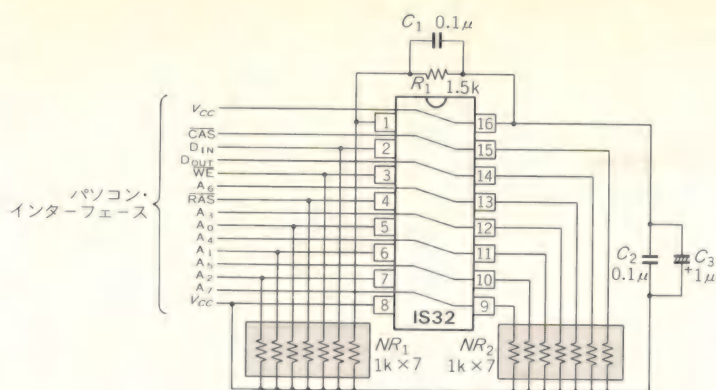
製作で留意する点は、レンズの取り付け部分とセンサ・ボード(IS32ボード)を機械的に一体化することがコツといえます。この部分は後で調整の対象となりますから、分解もできる構造でなければなりません。

ハウジング内部は暗く、光の反射が生じないようにします。

寸法的に留意する点は、レンズとセンサの距離です。



〈図8〉 センサ・ボード回路



今回使用したレンズはCマウント・タイプで、取り付けネジ終端のフランジと、センサの結像位置の間隔が17.526mmとなっています。

ほかのマウント形式でも決まっているはずで、これを確かめて守らなければなりません。これを誤ると、レンズを調整してもピントが合わない結果になります。

図8にセンサ・ボードの回路図を示します。電源を除く全端子を1kΩの抵抗でプルアップします。こうして回路インピーダンスを下げ、ケーブルの干渉を低くし、また素子の保護を兼ねます。

電源のパスコンはセラミック型を使って、DRAMの瞬時電流の影響を吸収します。ケーブルは直付けとし、末端にコネクタを配置しています。写真2に製作したカメラの外観を示します。

### ● パソコン・インターフェース回路

インターフェースは、撮像部をコントロールし、撮像データをCPUへ受け渡すことを目的としますが、この部分はCPU側、とりわけソフトウェアと深い関連をもちますから、設計に当たっては双方の分担をよく練ってかかる必要があります。

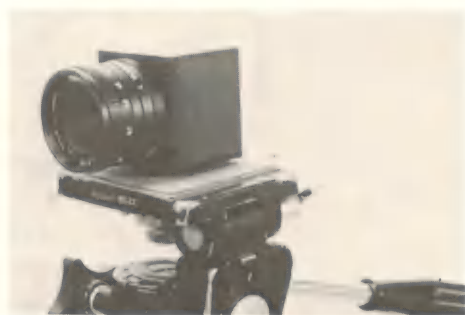
## カメラ・レンズの選び方

レンズの選択は重要です。特に撮像範囲に対し焦点距離を割り出す必要があります。これについてレンズの基礎的なことを復習しておきましょう。

光源からの光波は空間を球状に広がっていきます。図Aの凸レンズから一定の距離にある被写体ABからくる光線は屈折された後、それぞれA'B'に集結し、この空間にABと相似の形が逆さになって再現します。

ここへ、トレース用紙やスリガラスを光軸に直角におけば、その像を見ることができます。そしてこの位置が焦点位置となります。

ここでaが無限に遠くて、AやB点からくる光線がおおの平行になって入ってくるとき、像面と光軸



〈写真2〉 製作したデジタル・カメラ

回路図と主要部のタイムチャートを図9と図10に示します。

回路はポート・デコーダ、データ・ラッチ、IS32制御のタイミング信号の発生の三つからなっていますが簡素なもので、特に取り上げるほどのものではありません。

まず出力ポートD<sub>BH</sub>はリフレッシュを受けもちます。このポートへIS32の行アドレスを送ると、直ちにその行をリフレッシュするように働きます。ですから、

との交点をレンズの主焦点といい、そのときのbをレンズの焦点距離としています。

これは、そのレンズの一定値であるとともにbの最小値でもあるわけです。aが近づくとbは大きくなります。

レンズの焦点距離をfとすると、a、b、fの関係は次式で表されます。

$$(1/a) + (1/b) = (1/f) \dots\dots\dots(1)$$

被写体ABの大きさと、結像A'B'の大きさは、図Aよりaとbの比になることがわかります。したがって像の倍率mは、

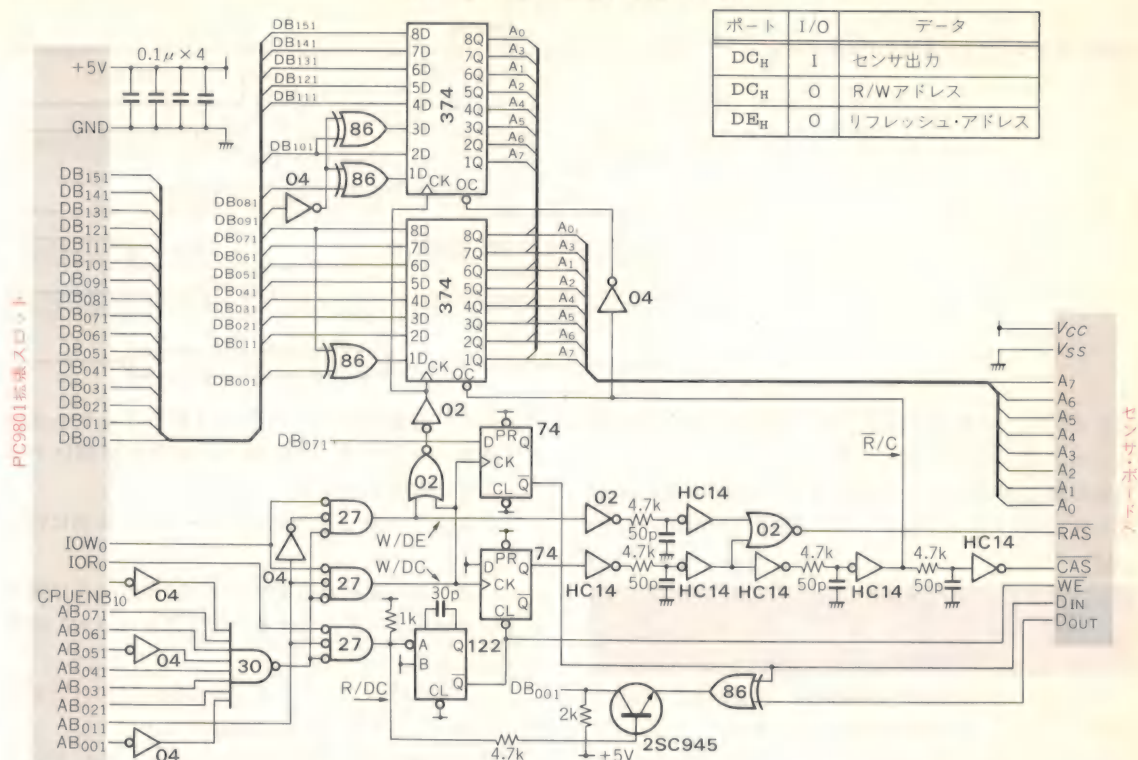
$$m = (A'B')/AB = b/a \dots\dots\dots(2)$$

となります。

本器で製作するときの問題を解いてみましょう。レ



〈図9〉 パソコン・インターフェース回路



適当な間隔でアドレスを更新しながら出力することで、メモリ内容を保持することができます。

出力ポートDCHは、読み出しアドレスをIS32へ渡すためのポートです。このポートへはワード単位(16ビット)でデータを送ります。MSDが列アドレス、LSDが行アドレスになります。行アドレスの最上位ビット(MSB)が“0”の場合が上部、“1”の場合が下部のセル・ゾーンを選択します。

このポートをコールすると、そのアドレス点のセル内容をIS32は出力しますから、CPU側はすぐに入力ポートDCHからデータを入力する必要があります。データは1ビットで、データ・バスのLSBへ乗ります。

また、このポートから入力した後、IS32の同一アドレスへ“1”が書き込まれるようになっています。

出力ポートDCHと入力ポートDCHはペアで動作

レンズの焦点距離 $f=16\text{ mm}$ で、撮像素子上へ実物の $1/50$ の大きさに映したいときは(2)式で、

$$m=b/a=1/50$$

$$\therefore b=a/50$$

$f=16$  ですから(1)式は、

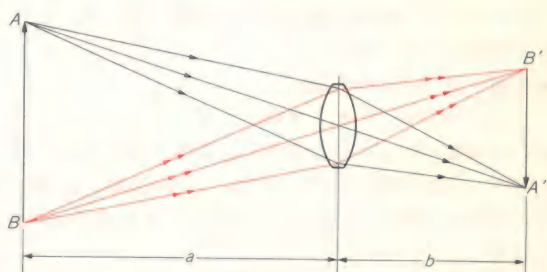
$$1/a+50/a=1/16$$

となり、これから、 $a=816$ 、 $b=16.32$  が求まります。つまり、被写体から $81.6\text{ cm}$ 離れた所にレンズがくるようにカメラをおいて、 $\infty$ のレンズ位置から、 $b-f=0.32\text{ mm}$ だけレンズを前進させれば、焦点が合うことになります。

以上は単純な式で、実際はレンズの厚みなどを計算に入れなければ精密な値は得られませんが、撮影距離が非常に近い場合をのぞき、実用的にはこれで十分な

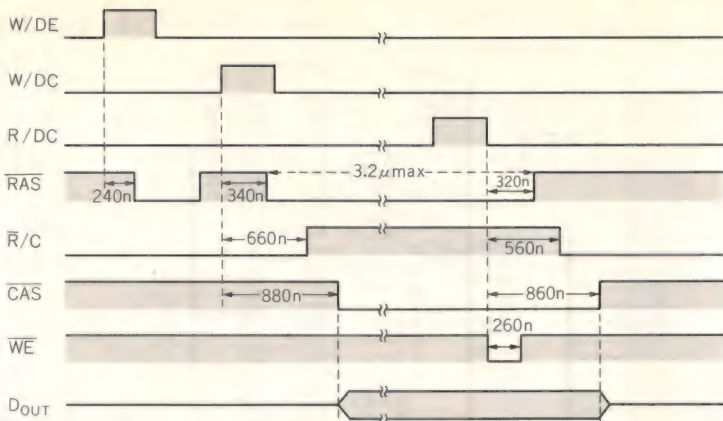
目安となります。表1(p.164)のセル・アレイの寸法をこの式に加えれば、被写体の寸法に対する撮像範囲が求まりますから、実状に合ったレンズの選択ができます。

〈図A〉 レンズの説明図





〈図10〉 インターフェース部タイムチャート



しますから、必ずDCHへ出力した後は、続けてDCHから入力しなければなりません。こうしないと、IS32のRASが下がったままになり、規定時間を越えるからです(タイムチャートの時間はこれを行った時の値)。

## ソフトウェア

プログラムの主要部は撮像制御、画像解析、画面コピー、ファイルの4部で構成しています。

ファイル部のみBASICで行い、ほかはアセンブラで作成しています。実験と評価を目的としたもので、このままでは実用的ではありませんが、若干の機能を追加することで、高い効果を発揮できます。

アセンブラ部の機能は、BASICプログラムで呼び出して使用します。アセンブラ部をリスト1、BASIC部をリスト2に示します(175ページより)。

### ● 撮像制御

今回はIS32の重要な制御となるリフレッシュをソフト指示で行うことで、ハードの複雑さから逃げる計画を立てました。そのためCPU 8086のI/Oタイミングを簡単に調べて目安を立ててスタートしましたが、実際に製作してやってみると、計画とは違った結果がでてあわてました。

これはPC9801のI/Oアクセスが、内部ハードウェアの構成法から遅くなっているようで、これが計画を狂わされる結果となりました。

プログラムの概要を順を追って説明しますが、リストを参考にしてください。

### ▶ 撮像パラメータの指定

プログラムをコールする前に、ポインタ“SPARA”へ撮像パラメータを書き込みます。値はバイト単位で0~255の範囲です。この値に25msを乗じた値がセルに対する露光タイムとなります。0を書くと256と見なします。

ポインタ“NP”へ“0”を入れるとセル・ゾーン

の上部だけ撮像を行い、“1”を入れると全セルに対して撮像が可能です。この場合は水平中央部にデッド・ゾーンが発生します。

“SPARA”と同じくプログラム・コールの前に行います。

プログラムの最初の主要なジョブは割り込みに関することです。リストの“タイマ割り込みセット”の部分が行います。

ここではIS32のリフレッシュを規則的に行うために、タイマ割り込みをセットして起動します。

タイマの繰り返し時間を約50μsにセットしています。こうして50μsごとに割り込みルーチンを実行させます。割り込みルーチンは、アドレス22EHにあり、呼び出されるごとに行アドレスをインクリメントし、出力ポートDEHへ送ります。行アドレスの変化範囲は0~FFHとしており、これはセル・アレイ2群に相当します。

したがって、1行に対するリフレッシュの繰り返し時間は、 $50\mu s \times 256 = 12.8ms$ となりますが、この時間は20msまで安全であることを確かめています。

### ▶ 撮像開始

撮像の準備として、全セルへ“1”の書き込みを行います。これがリストの“Clear IS32”のルーチンです。

“1”の書き込みが終わると直ちに撮像に移ります。“SOAK”がこのルーチンです。先の露光パラメータを元に露光タイムの期間割り込みを停止して、このルーチンをループします。

割り込みを停止すると、リフレッシュがとまり、セルの電荷が光の度合いに応じてディスチャージされます。露光時間を過ぎると割り込みが開始してリフレッシュが再開され、撮像パターンが定着することになります。

ここで重要な問題があります。IS32がDRAM構造であることから、チップ全体に活動を停止するとまずいのです。後で判明したのですが、停止するとリフレ



ッシュ再開時にデータが化けることがあります。

これはDRAM内部の動作要因にからむ問題で、メーカーからコメントがないので正確な原因は不明です。そこで、ダミー行を定め、撮像中もこの行だけはリフレッシュを続けることで、上記トラブルを解消できます。

#### ▶データ読み込みと補間

次がセル内容の読み出しですが、どの順序で行うかが問題です。

図4のセル配置を注意して見ると、最上行 $Y_0$ と最下行 $Y_{128}$ および左右端の各2列を除くエリアの各行は一定の変化を繰り返しています。さらに、1行とびごとにほとんど同一のパターンのならびであること、すなわち偶数と奇数行の2種のパターンに絞れることがわかります。

次に補間の問題です。この方法はいろいろ考えられますが、今回は図11のように上下行の近接したデータ間の論理積を行い、結果をスペース部分へ埋めていく方法としました。

これは一度には無理ですから、まず行ごとにスペース部分へ隣のデータをコピーしながら読み出していきます。これを行っているのが“データ読み取り、画面表示”のルーチンです。読み取りデータのバッファはグラフィック用のVRAMを使用していますから、すぐにCRTへ表示が行えますが、ここでの表示は補間された画像ではありません。

データ・リードが終わると、IS32の仕事は終わったわけですから、CPUの効率を上げるために、先に行

った割り込み手続きをキャンセルします。

最後の仕上げがデータ補間で、これはVRAM上のデータに対して直接行います。上行1バイトと下行1バイト間でANDをとり、これを上行へ戻しながら行います。

このとき大切なことは、下行の1バイトには、補間に必要のないビットにマスクをかけ、上行の有効ビットを壊さないようにして行うことです。このルーチンが“データ補間”です。これを終わってプログラムが終了します。

このカメラでの撮影のサンプル画像を写真3に示します。VRAMの内容をそのままハード・コピーしたものです。

#### ●画像解析

カメラで撮影した画像データは、コンピュータのグラフィック表示用メモリに収集されて、CRTに表示されます。

しかし、このままではテレビ・モニタと同様で、コンピュータに接続した意味をもちません。問題はコンピュータへ入力した画像データをいかに処理し、どう生かすかで、これからが本番といえます。

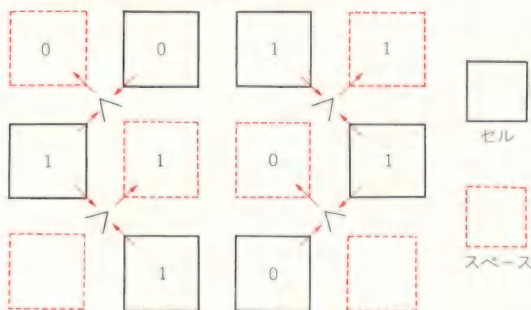
画像解析は目的が広汎であり、手法も単純なものから複雑なものまで多岐を極めています。

今回はあまり欲張らないで、1次的な処理を行うこととし、後の高次処理への基礎データ収集も兼ねることから、パターン分布を扱うことにしました。

写真4のようなランダムに分布したパターンを個々にサーチして、それぞれのサイズ(絵素数)と重心座標値を解析し、コンピュータのメモリ上へ配列するといふものです。

これだけの機能でも応用価値は大きく、散在量が限度を越えると目視での計数は困難です。しかも不整形な物の大きさの比較が大変であることは日常経験する

〈図11〉補間とは



## トランジスタ技術

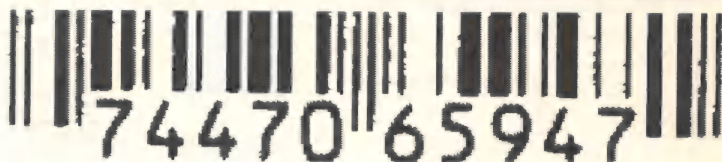
〈写真3〉  
撮像サンプル



原画像▲

撮像画像▶

## トランジスタ技術



〈写真4〉



パターン内部のサーチは、パターンの形状によって  
はかなりの工夫を要し、これの良し悪しで時間やメモ

SX, SYの値を〔MEMO〕と称するメモ領域へ格納します。SX, SYは、現在のサーチ対象のドット位



置を座標値で示すポイントです。

いよいよサーチ開始ですが、大事なことは、方向を定めて行うことで、この原則は必ず守る必要があります。ここでは左方向から行いますが、現在の位置はそのままにして、SX、SYの値を1だけ減じ(デクリメント)、サーチ位置を左上点へ移動し、この点のドットをテストします。

これはサーチ列を幹とすると、この上下行は枝と見なせますから、枝の有無を知ることになります。

この位置が“1”であれば枝が存在し、これが前の枝に連らなったものであるかをフラグUFの状態で判定します。UFの値が“0”のときは、前に枝は存在せず新しく枝が発生したものとします。そして、この位置をスタックに格納しますが、その前にSPの値をデクリメント(スタックを1段下げる)して行います。

ここでUFへ“1”をロードし枝の発生を知らせます。枝が生じないときはUFへ“0”をロードし、次の処理へブランチします。枝が存在してもUFが“1”のときは、枝が連らなっているものとし、そのまま次へブランチします。

次は反対面(下側)のテストを行います。これは上部と同様でフラグをLFに変え、SYへ2をプラスして行います。

● 行列の処理

以上で上下行のドット・テストを終え、次はSYをデクリメントして、対象行列の処理です。

この点のドットが“1”であれば、CN(カウンタ)に1をプラス(インクリメント)し、ドットに対して処理済みのマーキングを行います。

ここでは“0”を該当ドットへロードしています。この後は再び座標を左へ移動しながら、今までの処理を繰り返します。

対象行列のドットが“0”になったときは、左方向

にパターンの切れ目が生じたときですから、左方向サーチを中止し、右方向サーチにチェンジします。

先に〔MEMO〕に格納した内容を再びSX、SYへ移し、UF、LFへ“0”をロードして右方向列サーチの準備とします。

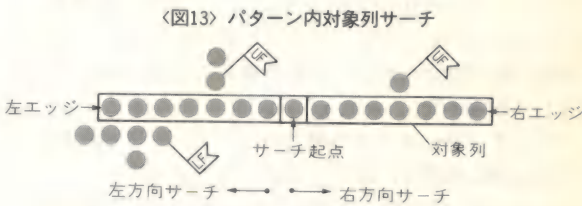
今のSX、SY位置のドットは“1”に決まっていますから、CNをインクリメントし、該当ドットへ“0”をロードします。右方向のサーチも左方向と類似していますから、図を参考にしてください。

右方向のサーチが終わるのは、対象列の右端に有効ドットが消えたときで、この時点で対象行列の処理を終えます。図13に列サーチの概略を示します。

ここまでのところで大事なことは、枝が発生しても、この位置をスタックに納めるだけにして処理は行わないことです。これを行うと、複雑なパターンでは枝が際限なく発生して、元の位置を見失ない、ついには迷子になってしまいます。

さて、ここで次の対象行列の起点を見つけなければなりません。

これはスタックの内容を取り出せばよいのです。その前にスタックの位置が上限値より下にあるかを確認めます。この位置が上限値であるときはスタックは空であり、対象パターンのサーチを終了したことになります。この時のCNの内容が、パターン構成のドット数となります。



〈図12〉 パターン内サーチ(つづき)

略号、記号	略号名	動作概要
SP	スタック・ポイント	棚状に配置されたメモリの現在位置を示すポイント。初期値は棚の上限値を示す。データを蓄わえる際、SP値をデクリメントし、その位置へデータをストアする。データを読み取る際は、SP値の示す位置から行い、その都度SP値をインクリメントする。
SX	X軸サーチ・ポイント	X軸の座標値を示す。
SY	Y軸サーチ・ポイント	Y軸の座標値を示す。
UF	アップ・フラグ	サーチ列の上行に有効ドットを発見した時に立てる(“1”をロード)。有効ドットが消えた点で、降ろす(“0”をロード)。
LF	ロワー・フラグ	サーチ列の下行の変化に対応し、扱いは、UFと同じ。
CN	カウンタ	有効ドット数のカウントに用いる。Job終了時の値が対象パターンを構成する総ドット数となる。
MEMO	メモ	一時記憶のためのバッファ・メモリ。
[ ]		[ ]内の値で示される位置のデータ値を示す。
←		右辺の内容を左辺へコピー。
=		両辺の内容を比較する。

(C) フローチャートの見方



スタック位置が上限値より下のときは、SX, SYへスタック内容を読み取り、これを新しい対象行列の起点として、フローチャートの④点へ戻ります。そして前と同一手順のサーチを繰り返します。

実際のプログラムはリスト1の“画像解析”で行い、ここではドット計数だけでなく、有効ドットの発生ごとにその位置(SX, SY)の値を積算できる機能をもたせています。

これは、後で個々のパターン・ブロックに固有の重心座標値を得るためです。この重心というのは、パターンを各点の質量が一定の物理的集合体と見なして、これに対する重量のバランス点としているからです。真円の場合は円中心となります。

この値は画像解析では重要なパラメータとなり、形状比較や認識などの高次処理でも必要となるものです。

これを求めるには、重心点座標を $\bar{X}$ ,  $\bar{Y}$ とすると、

$$\bar{X}, \bar{Y} = \frac{\sum X, Y}{n} \quad n = \text{ドット分布数}$$

で求めることができます。

これより、パターンの重心点座標は、パターン構成の全ドット座標の平均値であることがわかります。

今回のプログラムでは行っていませんが、サーチの途中、パターン・エッジも認識できますから、輪郭の抽出が可能です。これを座標変換することでパターンの加工が容易になります。

本ルーチンはBASICから呼び出して使用しますが、呼び出しの前にデータ・エリアを確保するために、2次元の配列宣言を行い、この領域の位置を渡すようにして行います。

ルーチンは、サーチ結果であるパターンごとのドット数と重心座標を、配列エリアへ戻します。

このようにBASICと往来できるプログラム構造にしておくと、後のシミュレーションなどに便利で、機能拡張を効率的に行うことができます。最終段階でアセンブラへ落とせばよいのです。

プログラムの詳細はリスト1、リスト2とフローチャートを参考してください。

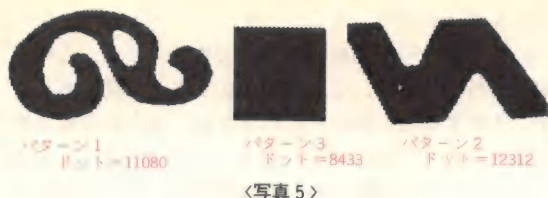
“画面コピー”と“ファイル”はテーマから外れますから割愛します。

なお、本プログラムの著作権などは(株)デジック・システムズに帰属します。実験などの個人レベルの利用には供しますが、営利対象への使用、引用は控えてください。

### ● 不整形パターン面積の解析

本カメラに解析用ソフトが加わることで、通常のカメラとは異なった方向での利用が可能になります。

まだ機能が不足で、高度な応用には適しませんが、応用上のヒントとして、不整形パターンの面積を求める例を紹介します。



整形パターンについては各部の寸法を測れば簡単に算出できますが、写真の左と右のパターンのように、不整形なものでは簡単にはいきません。

写真5は、黒い紙から無造作に切り出したものをならべ、本カメラで撮影したものです。中央の正方形のパターンは寸法がわかっていて、各辺が60mmで、面積は3600mm<sup>2</sup>となっています。これを解析プログラムにかけるとどうなるでしょう。

たちどころに写真の下部に示すような数値が戻ってきます。

プログラムは画面左上からサーチを開始し、はじめに見つかったパターン1から構成ドット数を計数し、続けて2, 3のパターンについても同様に行います。

それぞれのドット数と重心座標値は、BASICの配列に格納されていますから、簡単なプログラムでパターン番号を指定すると、即座にデータを読み取ることができます。

このデータから面積を求めるのは簡単です。正方形の面積は出ていますから、このドット数でほかのパターンのドット数を割ります。こうすることで、正方形を1とした各パターンとの面積比が求まります。

この比に正方形の面積を掛けた値が対象パターンの面積になります。

計算結果はパターン1が4730mm<sup>2</sup>、パターン2が5256mm<sup>2</sup>となります。

この式をプログラムに組み込んで面積と指定すれば即表示できますので、実用的になるでしょう。

このように、既知のパターンを置かなくても、撮影範囲が定まっていれば、この寸法で全体の面積を求め、これを基準値として行えばよいのです。

パターン数は多くてもよく、今回は配列サイズを1000としていますから、1000個のパターンを対象にできます。しかし、あまり欲張ると精度が落ちて実用的ではありません。実用精度との関係を実際に見極めて行う必要があります。

先の写真4は、各種のネジ類を散在させて撮影したもので、全部のパターン数の解析値は332で、おのこのパターンの面積と重心座標の収集を、わずか1.7秒ですませます。

このようなパターンの場合、サイズのレベルを数段階に区切って、それぞれの境界内に含まれるパターンの個数を引き出して処理することで、サイズ別の勢力



分布や傾向を知ることができます。

このほか、解析データを元に外部機器の制御などへ機能拡張すれば、より効果の大きい応用の道が開けることになります。

## ● おわりに

カメラでの画像解析の良さは、対象物に非接触で行えること、そしてレンズの選択や操作次第でマクロか

らミクロまで広い範囲を対象にできることでしょう。

このように簡易な入力装置でも、収集データの処理次第で相当な力をもち得ることがわかりでしょう。

## ◆参考·引用\*文献◆

- (1)\*IS32, オプティックRAMデータ・シート, マイクロン・テクノロジー.  
(2) デジック, パーソナル・スコープISC-100取り扱い説明書.

## &lt;リスト1&gt;

"ISC-50" 報告制度, データ解析用プログラム  
Copyright 1986, 7 Digic Systems Corp.

```
;;Input: [SPARA] Soak parameter (0 to 255)
::      [NP]    C.Zone select (0=Half 1=All)
```

0000	ISC	SEGMENT ASSUME	CS:ISC,DS:ISC,ES:ISC,SS:ISC
= A800	GVRAM	EGU	0A800H
= 0002	IMR	EGU	02H
= 0000	E01	EGU	00H
= 00DE	REFADR	EGU	0DEH
= 00DC	DATADR	EGU	0DCH
= 00DC	ISDATA	EGU	0DCH
= 0008	INTVEC	EGU	08H
= 007B	CUN10M	EGU	123
= 0064	CUN8M	EGU	100
= 3110	STP286	EGU	12560
= 1E3C	STP10M	EGU	7740
= 17FC	STP8M	EGU	6140
= 0F1E	STP5M	EGU	3870
= 0077	TIMECW	EGU	77H
= 0071	TIMEQ	EGU	71H
= 0000	DEADRO	EGU	0
= 2DA0	DEADR1	EGU	80*146

```
0000      ORG      00H
          :////// Program Start here //
```

START:	JMP	PHOTO	映像制御
0000	EB 09 90		
0000	E9 0249 R	ANALYZ	画像解析
0006	E9 063F R	GCOPY	画面コピー
0009 ??	DB	?	？ 藍光パラメータ
000A 00	DB	0	？ ゼルソーン指定
	PAGE		

```

000B 1E
000C 00
000D 50
000E 50
000F 53
0010 51
0011 52
0012 E4
0013 DG
0014 8C
0015 C8
0016 E8
0017 D8
0018 1E
0019 33
001A C0
001B D8
001C A0
001D 50
001E 1F
001F A8
0020 80
0021 BB
0022 17FC
0023 75
0024 0A
0025 A8
0026 0A
0027 0A
0028 0A
0029 BB
002A 0F1E
002B 74
002C 03

```



002E BB 1E3C	MOV	EX,STP10M	00AE 75 F5	JNZ	XS1
0031 24 F0	AND	AL,OF0H	00B0 FB	STI	
0033 3C A0	CMP	AL,OA0H			
0035 B8 9090	MOV	AX,9090H			
0038 75 06	JNZ	SIX			
003A B8 5850	MOV	AX,5850H			
003D BB 3110	MOV	EX,STP286			
0040 2E: A3 0097 R	MOV WORD PTR CS:[CHG1],AX				
0042 2E: A3 0220 R	MOV WORD PTR CS:[CHG2],AX				
0048 2E: 89 1E 00A6 R	MOV WORD PTR CS:[XS1+1],BX				
004D FA	CLI				
004E 1E	PUSH	DS			
004F 33 C0	XOR	AX,AX			
0051 8E D8	MOV	DS,AX			
0053 BE 0008	MOV	SI,INTVEC			
0056 D1 E6	SHL	SI,1			
0058 D1 E6	SHL	SI,1			
005A B8 022E R	MOV	AX,OFFSET REFRSH			
005D 8C CB	MOV	EX,CS			
005F 87 04	XCHG	AX,[SI]			
0061 87 5C 02	XCHG	EX,[SI+2]			
0064 8A 16 0501	MOV	DL,DS:[501H]			
0068 1F	POP	DS			
0069 A3 0767 R	MOV	[INTOFF],AX			
006C 89 1E 0769 R	MOV	[INTSEG],BX			
0070 B0 34	MOV	AL,00110100B			
0072 E6 77	OUT	TIMECW,AL			
0074 F6 C2 80	TEST	DL,80H			
0077 B8 0064	MOV	AX,CUN8M			
007A 75 03	JNZ	ST1			
007C B8 007B	MOV	AX,CUN10M			
007F E6 71	OUT	TIME0,AL			
0081 8A C4	MOV	AL,AH			
0083 E6 71	OUT	TIME0,AL			
0085 90	NOP				
0086 E4 02	IN	AL,IMR			
0088 A0 076B R	MOV	[IMRBF],AL			
008B B0 FE	MOV	AL,11111110B			
008D E6 02	OUT	IMR,AL			
008F FB	STI				
0090 33 C9	XOR	CX,CX			
0092 8B C1	MOV	AX,CX			
0094 FA	CLI				
0095 E7 DC	OUT	DATADR,AX			
0097 90	NOP				
0098 90	NOP				
0099 E4 DC	IN	AL,ISDATA			
009B FE	STI				
009C E2 F4	LOOP	CL1			
009E FA	CLI				
009F 8A 1E 0009 R	MOV	BL,[SPARA]			
00A3 B0 FF	MOV	AL,OFFH			
00A5 B9 0F1E	MOV	CX,STP5M			
00A8 E6 DE	OUT	REFADR,AL			
00AA E2 FC	LOOP	BL			
00AC FC	DEC	BL			
00B0 3C A0	CMP	AL,OA0H			
00B2 B8 A800	MOV	AX,GVRAM			
00B4 8E D8	MOV	DS,AX			
00B6 BE 0000	MOV	SI,DFADRO			
00B8 B2 01	MOV	DL,01			
00BA B7 40	MOV	BH,64			
00BC B7 40	MOV	BH,64			
00BE E8 011C R	DSPUPP				
00C1 2E: 80 3E 000A R 00	CALL	BYTE PTR CS:[NP],0			
00C7 74 0A	JZ	RESINT			
00C9 BE 2DA0	MOV	SI,DFADR1			
00CB B2 80	MOV	DL,128			
00CD B7 40	MOV	BH,64			
00D0 E8 017F R	CALL	DSFLOW			
00D3 FA	CLI				
00D4 1F	POP	DS			
00D5 A1 0767 R	MOV	AX,[INTOFF]			
00D8 8B 1E 0769 R	MOV	EX,[INTSEG]			
00DC 1E	PUSH	DS			
00DD 50	PUSH	AX			
00DE 33 C0	XOR	AX,AX			
00E0 8E D8	MOV	DS,AX			
00E2 58	POP	AX			
00E3 BE 0008	MOV	SI,INTVEC			
00E6 D1 EE	SHR	SI,1			
00E8 D1 EE	SHR	SI,1			
00EA 89 04	MOV	[SI],AX			
00EC 89 5C 02	MOV	[SI+2],BX			
00EF 1F	POP	DS			
00F0 A0 076B R	MOV	AL,[IMRBF]			
00F3 E6 02	OUT	IMR,AL			
00F5 FB	STI				
00F6 B8 A800	MOV	AX,GVRAM			
00F9 8E D8	MOV	DS,AX			
00FB BE 0000	MOV	SI,DFADRO			
00FE BF 0050	MOV	DI,DFADRO+80			
0101 E8 01E2 R	CALL	REV DAT			
0104 2E: 80 3E 000A R 00	CALL	BYTE PTR CS:[NP],0			
010A 74 09	JZ	EXIT			
010C BE 2DA0	MOV	SI,DFADR1			
010F BF 2DF0	MOV	DI,DFADR1+80			
0112 E8 01E2 R	CALL	REV DAT			
0115 5A	EXIT:	POP	DX		
0116 59	POP	CX			
0117 5B	POP	BX			
0118 58	POP	AX			
0119 5E	POP	SI			
011A 1F	POP	DS			
011B CF	IRET				
011C B6 00	DSPUPP:	MOV	DH,00		
011E B3 3F	EVEN:	MOV	CH,63		
0120 E8 021B R	EV1:	CALL	REDDAT		
0123 FE CA	DEC	DEC	DL		



0125	FE C6	INC	INC	FE C2	INC	DL
0127	FE C6	INC	DH	01B2	MOV	CH,63
0129	E8 021B R	INC	REDDAT	01B4	MOV	DH,01
012C	FE C2	INC	CALL	01B6	MOV	REDDAT
012E	E8 021B R	INC	REDDAT	01B8	DEC	DL
0131	FE CA	DEC	DL	01BB	CALL	REDDAT
0133	FE C6	INC	DH	01BD	CALL	REDDAT
0135	FE C6	INC	DH	01C0	INC	DL
0137	E8 021B R	INC	REDDAT	01C2	INC	DH
013A	FE C2	INC	DL	01C4	INC	DH
013C	88 0C	MOV	[SI],CL	01C6	CALL	REDDAT
013E	46	INC	SI	01CB	CALL	REDDAT
013F	FE CD	DEC	CH	01CE	INC	DL
0141	75 DD	JNZ	EV1	01D0	INC	DL
0143	B9 0011	MOV	CX,17	01D2	INC	DH
0146	03 F1	ADD	SI,CX	01D4	INC	DH
0148	FE CF	DEC	BH	01D6	MOV	[SI],CL
014A	75 01	JNZ	ODD	01D7	INC	SI
014C	C3	RET		01D9	DEC	CH
014D	B5 3F	ODD:	CH,63	01DB	JNZ	LOD1
014F	B6 01	MOV	DH,01	01DD	MOV	CX,17
0151	E8 021B R	CALL	REDDAT	01DE	ADD	SI,CX
0154	FE C2	INC	DL	01E0	JMP	DSFLOW
0156	E8 021B R	CALL	REDDAT			
0159	FE CA	DEC	DL	01E2	REV DAT:	
015B	FE C6	INC	DH	01E2	MOV	BH,63
015D	FE C6	INC	DH	01E4	MOV	CX,63
015F	E8 021B R	CALL	REDDAT	01E7	MOV	AL,[DI]
0162	FE C2	INC	DL	01E9	REVEVN:	OR
0164	E8 021B R	CALL	REDDAT	01EB	AND	AL,01100110B
0167	FE CA	DEC	DL	01ED	INC	[SI],AL
0169	FE C6	INC	DH	01EE	INC	SI
016B	FE C6	INC	DH	01EF	INC	DI
016D	88 0C	INC	[SI],CL	01F1	LOOP	REVEVN
016F	46	INC	SI	01F4	MOV	CX,17
0170	FE CD	DEC	CH	01F6	ADD	SI,CX
0172	75 DD	JNZ	OD1	01F8	ADD	DI,CX
0174	FE C2	INC	DL	01FB	MOV	CX,63
0176	FE C2	INC	DL	01FD	MOV	AL,[DI]
0178	B9 0011	MOV	CX,17	01FF	AND	SI,AL
017B	03 F1	ADD	SI,CX	0201	INC	DI
017D	EB 9D	JMP	DSPUUP	0202	INC	DI
017F	B6 00	DSFLOW:	DH,00	0203	LOOP	REVODD
0181	B5 3F	MOV	CH,63	0205	MOV	CX,17
0183	E8 021B R	CALL	REDDAT	0208	ADD	SI,CX
0186	FE C2	INC	DL	020A	ADD	DI,CX
0188	FE C6	INC	DH	020C	DEC	BH
018A	FE C6	INC	DH	020E	JNZ	REV1
018C	E8 021B R	INC	REDDAT	0210	MOV	CX,63
018F	FE CA	CALL	REDDAT	0213	XOR	AL,AL
0191	E8 021B R	CALL	REDDAT	0215	MOV	[SI],AL
0194	FE C2	INC	DL	0217	INC	SI
0196	FE C6	INC	DH	0218	LOOP	REVX
0198	FE C6	INC	DH	021A	RET	
019A	E8 021B R	CALL	REDDAT	021B	REDDAT:	MOV
019D	FE CA	DEC	DL	021D	CLI	AX,DX
019F	88 0C	MOV	[SI],CL	021E	OUT	DATADR,AX
01A1	46	INC	SI	0220	NOP	
01A2	FE CD	DEC	CH	0221	NOP	
01A4	75 DD	JNZ	LEV1	0222	IN	AL,ISDATA
01A6	B9 0011	MOV	CX,17	0224	STI	AL,1
01A9	03 F1	ADD	SI,CX	0225	ROR	CL,1
01AB	FE CF	DEC	BH	0226	RCL	AL,1
01AD	75 01	JNZ	LODD	0228	RCL	AL,1
01AF	C3	RET		0229	RCL	CL,1
01B0	FE C2	LODD:	INC	022D	RET	



022E 50	0293 B9 003F	TRS1X: MOV CX,63
022F 8A C3	0296 AC	TRS2: LODSB
0231 E6 DE	0297 32 C4	XOR AL,AH
0233 FE C3	0299 AA	STOSB
0235 B0 20	029A E2 FA	LOOP
0237 E6 00	029C 83 C6 11	ADD SI,17
0239 58	029F 83 C7 11	ADD DI,17
023A CF	02A2 4B	DEC BX
	02A3 75 EE	TRISX JNZ
	02A5 B8 B000	MOV AX,GHSEG+800H
	02A8 8E D8	MOV DS,AX
	02AA 07	ES
	02AB 5F	POP DI
===== 制込み ルーチン ===== Refresh		
REFRSH: PUSH AX		
MOV AL,BL		
OUT REFADR,AL		
INC BL		
MOV AL,20H		
OUT EOI,AL		
POP AX		
IRET		
PAGE		
*****		
***** 画像更新 *****		
GHSEG EQU 0A800H		
SX EQU 0	02AC 26: C7 05 0000	INIT: MOV WORD PTR ES:[DI],0
SY EQU 0	02B1 2E: C7 06 023D R 0000	MOV CS:[XP],SX
EX EQU 503	02B8 2E: C7 06 023F R 0000	MOV CS:[YP],SY
EY EQU 125	02BF 2E: C6 06 023B R 00	MOV CS:[UF],0
	02C5 2E: C6 06 023C R 00	MOV CS:[LF],0
	02CB BD 2770 R	MOV BP,OFFSET STK
UF DB ?	02CE 2E: 8B 0E 023D R	INIT2: MOV CX,CS:[XP]
LF DB ?	02D3 2E: 8B 16 023F R	MOV DX,CS:[YP]
XP DB ?	02D8 33 DB	XOR BX,BX
YP DW ?	02DA 2E: 89 1E 0241 R	MOV CS:[XSUM],BX
XSUM DW ?	02DF 2E: 89 1E 0243 R	MOV CS:[XXSUM],BX
XXSUM DW ?	02E4 2E: 89 1E 0245 R	MOV CS:[YSUM],BX
YSUM DW ?	02E9 2E: 89 1E 0247 R	MOV CS:[YYSUM],BX
YYSUM DW ?	02EE 2E: 88 1E 023B R	MOV CS:[UF],BL
	02F3 2E: 88 1E 023C R	MOV CS:[LF],BL
	02F8 EB 0624 R	CALL ?XYADR
ANALYZ: PUSH AX	02FB 84 04	RSRC: TEST [SI],AL
PUSH EX	02FD 75 45	JNZ RSRC2
PUSH CX	02FE 81 F9 01F7	CMP CX,EX
PUSH DX	0303 74 0A	JZ RSRC1
PUSH BP	0305 41	INC CX
PUSH SI	0306 D0 E8	SHR AL,1
PUSH DI	0308 73 F1	JNC RSRC
PUSH ES	030A B0 80	MOV AL,80H
PUSH DS	030C 46	INC SI
MOV ES,[BX+6]	030D EB EC	JMP RSRC
MOV DI,[BX+4]	030F 83 FA 7D	CMP DX,EY
MOV AX,ES:[DI]	0312 74 26	JZ RSEXT
MOV ES,[BX+2]	0314 42	INC DX
MOV DI,[BX]	0315 B9 0000	INC CX
MOV DI,ES:[DI]	0318 E8 0624 R	CALL ?XYADR
MOV ES,AX	031B EB DE	CALL RSRC
MOV AX,GHSEG	031D 2E: C7 05 FFFF	EMEXT: MOV WORD PTR ES:[DI],-1
MOV DS,AX	0322 BB 007E	SEEXT: MOV BX,126
CLD	0325 B8 B000	MOV AX,GVGRAM+800H
WORD PTR ES:[DI],0	0328 8E C0	MOV ES,AX
PUSH DI	032A BF 0000	MOV DI,0
PUSH ES	032D 32 C0	XOR AL,AL
PUSH DS	032F B9 003F	MOV CX,63
POP ES	0332 F3/ AA	REP STOSB
POP DS	0334 83 C7 11	ADD DI,17
MOV BX,126	0337 4B	DEC BX
MOV DI,8000H	0338 75 F5	JNZ EME1
TRISX JZ	033A 1F	POP DS
MOV CX,63	033B 07	POP ES
REP MOVSB	033C 5F	POP DI
DEC CX	033D 5E	POP SI
JZ TRS3	033E 5D	POP BP
TRS3 JZ	033F 5A	POP DX
ADD SI,17	0340 59	POP CX
ADD DI,17	0341 5B	POP EX
JMP TRS0	0342 58	POP AX
JMP TRS3	0343 CF	IRET
MOV AH,OFFH		
TRIS1: MOV		
0249 50		
024A 53		
024B 51		
024C 52		
024D 55		
024E 56		
024F 57		
0250 06		
0251 1E		
0252 8E 47 06		
0255 8B 7F 04		
0258 26: 8B 05		
025B 8E 47 02		
025E 8B 3F		
0260 26: 8B 3D		
0263 8E C0		
0265 B8 A800		
0268 8E D8		
026A FC		
026B 26: 83 3D 00		
026F 57		
0270 06		
0271 1E		
0272 07		
0273 BB 007E		
0276 BE 0000		
0279 BF 8000		
027C 74 13		
027E B9 003F		
0281 F3/ A4		
0283 4B		
0284 74 1F		
0286 83 C6 11		
0289 83 C7 11		
028B 58 C0		
028E BF 15 90		
0291 B4 FF		



＜リスト1＞ つぎ

0344	MOV	RSRC2:	MOV	CS:[XP],CX	0401	75 09	JNZ	ULS7
0349	MOV		MOV	CS:[YF],DX	0403	2E: C6 06 023B R 00	MOV	CS:[UF],0
		ULSRC:	XOR	ISI,AL	0409	EB 24 90	JMP	ULS8
0350	ADD		ADD	CS:[XSUM],CX	040C	2E: 80 3E 023B R 00	CMP	CS:[UF],0
0355	JNC		JNC	ULS5X	0412	75 1B	JNZ	ULS8
0357	TNC		TNC	CS:[XSUM]	0414	4A	DEC	DX
035C	ADD	ULSR1:	ADD	CS:[XSUM],DX	0415	2E: 89 4E FE	MOV	CS:[BP-2],CX
0361	JNC		JNC	ULSR2	0419	2E: 89 56 FC	MOV	CS:[BP-4],DX
0363	JNC		JNC	CS:[YVSUM]	0420	81 FD 0770 R	SUB	BP,4
0368	JNC	ULSR2:	JNC	CX,0	0424	73 03	CMP	BP,OFFSET EMG
0369	CMP		CMP	CX,0	0426	E9 031D R	JNC	ULS7X
036C	JNZ		JNZ	ULS0	0429	2E: FE 06 023B R	JMP	EMEXT
036E	JMP		JMP	ULS6	042E	42	INC	CS:[UF]
0371	PUSH	ULS0:	PUSH	SI	042F	83 C6 50	INC	DX
0372	MOV		MOV	AH,AL	0432	83 FA 7D	ADD	SI,80
0374	DO E0		SHL	AL,1	0435	74 36	CMP	DX,EY
0376	JNC		JNC	ULS1	0437	83 C6 50	JZ	ULS12
0378	MOV		MOV	AL,1	043A	84 04	ADD	SI,80
037A	DEC		DEC	SI	043C	75 09	TEST	[SI],AL
037B	83 FA 00	ULS1:	CMP	DX,SY	043E	2E: C6 06 023C R 00	JNZ	ULS10
037E	74 39		JZ	ULS4	0444	EB 24 90	MOV	CS:[LF],0
0380	83 EE 50		SUB	SI,80	0447	2E: 80 3E 023C R 00	JMP	ULS11
0383	84 04		TEST	[SI],AL	044D	75 1B	CMP	CS:[LF],0
0385	75 09		JNZ	ULS2	044F	42	JNZ	ULS11
0387	2E: C6 06 023B R 00		MOV	CS:[UF],0	0450	2E: 89 4E FE	INC	DX
038D	EB 27 90	JMP	ULS3	ULS3	0454	2E: 89 56 FC	MOV	CS:[BP-2],CX
0390	2E: 80 3E 023B R 00	ULS2:	CMP	CS:[UF],0	0458	83 ED 04	MOV	CS:[BP-4],DX
0396	75 1E		JNZ	ULS3	045B	81 FD 0770 R	SUB	BP,4
0398	49		DEC	CX	045F	73 03	CMP	BP,OFFSET EMG
0399	4A		DEC	DX	0461	E9 031D R	JNC	ULS10X
039A	2E: 89 4E FE		MOV	CS:[BP-2],CX	0464	4A	JMP	EMEXT
039E	2E: 89 56 FC		MOV	CS:[BP-4],DX	0465	2E: FE 06 023C R	DEC	DX
03A2	83 ED 04	SUB	BP,4	BP,4	046A	83 EE 50	INC	CS:[LF]
03A5	81 FD 0770 R	CMP	BP,OFFSET EMG	ULS2X	046D	81 F9 01F7	SUB	SI,80
03A9	73 04	JNC	SI	ULS2X	0471	75 03	CX,EX	ULS12:
03AB	5E	POP	SI	SI	0473	E9 0504 R	JNZ	ULS12X
03AC	E9 031D R	JMP	EMEXT	EMEXT	0476	41	JMP	ULS18
03AF	41	ULS2X:	INC	CX	0477	DO E8	INC	CX
03B0	42		INC	DX	0479	73 03	SHR	AL,1
03B1	2E: FE 06 023B R		INC	CS:[UF]	047B	B0 80	JNC	ULS13
03B6	83 C6 50	ULS3:	ADD	SI,80	047D	46	MOV	AL,80H
03B9	83 FA 7D	ULS4:	DX,EY	ULS5X	047E	84 04	INC	SI
03BC	74 36		JZ	ULS5X	0480	74 1E	JZ	ULS14
03BE	83 C6 50		ADD	SI,80	0482	43	INC	EX
03C1	84 04		TEST	[SI],AL	0483	30 04	XOR	[SI],AL
03C3	75 09	JNZ	ULS5	ULS5	0485	2E: 01 0E 0241 R	ADD	CS:[XSUM],CX
03C5	2E: C6 06 023C R 00		MOV	CS:[LF],0	048A	73 05	INC	ULS3
03CB	EB 27 90	JMP	ULS5X	ULS5X	048C	2E: FF 06 0243 R	JNC	CS:[XSUM]
03CE	2E: 80 3E 023C R 00	ULS5:	CMP	CS:[LF],0	0491	2E: 01 16 0245 R	ADD	CS:[VSUM],DX
03D4	75 1E		JNZ	ULS5X	0496	73 05	JNC	ULS4
03D6	49		DEC	CX	0498	2E: FF 06 0247 R	INC	CS:[VVSUM]
03D7	42		INC	DX	049D	E9 03F7 R	JMP	ULS6
03D8	2E: 89 4E FE		MOV	CS:[BP-2],CX	04A0	83 FA 00	CMP	DX,SY
03DC	2E: 89 56 FC	MOV	CS:[BP-4],DX	CS:[BP-4],DX	04A3	74 2D	JZ	ULS16
03E0	83 ED 04	SUB	BP,4	BP,4	04A5	2E: 80 3E 023B R 00	CMP	CS:[UF],0
03E3	81 FD 0770 R	CMP	BP,OFFSET EMG	ULS5X	04AB	75 25	JNZ	ULS16
03E7	73 04	JNC	SI	SI	04AD	83 EE 50	SUB	SI,80
03E9	5E	POP	SI	EMEXT	04B2	84 04	TEST	[SI],AL
03EA	E9 031D R	JMP	EMEXT	CX	04B4	4A	JZ	ULS15
03ED	41	ULS5X:	INC	DX	04B5	2E: 89 4E FE	DEC	DX
03EE	4A		INC	CS:[LF]	04B9	2E: 89 56 FC	MOV	CS:[BP-2],CX
03EF	2E: FE 06 023C R	INC	INC	CS:[UF]	04BD	83 ED 04	MOV	CS:[BP-4],DX
03F4	5E	POP	SI	SI	04C0	81 FD 0770 R	SUB	BP,4
03F5	8A C4	MOV	AL,AH	AL,AH	04C0	81 FD 0770 R	CMP	BP,OFFSET EMG
03F7	83 FA 00	CMP	DX,SY	DX,SY	04C4	73 03	JNC	ULS14X
03FA	74 36	JZ	ULS9	ULS9	04C6	E9 031D R	JMP	EMEXT
03FC	83 EE 50	SUB	SI,80	SI,80	04C9	42	INC	DX
03FF	84 04	TEST	[SI],AL	[SI],AL	04CA	2E: FE 06 023B R	INC	CS:[UF]
					04CF	83 C6 50	ADD	SI,80



0402	83 FA 7D	ULS16:	CMP	DX,EY	ULS18	0594	2E: 80 3E 023B R 00	LSR2:	CMP	CS:[UF],0
0405	74 2D		JZ	ULS18	059A	75 1C			JNZ	LSR3
0407	2E: 80 3E 023C R 00		CMP	CS:[LF],0	059C	2E: 89 4E FE			MOV	CS:[BP-2],CX
0409	75 25		JNZ	ULS18	059C	2E: 89 56 FC			MOV	CS:[BP-4],DX
040F	83 C6 50		ADD	ULS18	05A0	2E: 89 56 FC			MOV	BP,4
0412	84 04		TEST	[SI],AL	05A4	83 ED 04			SUB	BP,4
0414	74 1B		JZ	ULS17	05A7	81 FD 0770 R			CMP	BP,OFFSET EMG
0416	42		INC	DX	05AB	73 06			JNC	LSR2X
0417	2E: 89 4E FE		MOV	CS:[BP-2],CX	05AD	5E			POP	SI
0418	2E: 89 56 FC		MOV	CS:[BP-4],DX	05AE	5E			POP	SI
0419	83 ED 04		SUB	BP,4	05AF	5E			POP	SI
0421	81 FD 0770 R		CMP	BP,OFFSET EMG	05B0	05B0			JMP	EMEXT
0422	73 03		JNC	ULS16X	05B3	2E: FE 06 023B R		LSR2X:	INC	CS:[UF]
0424	E9 031D R		JMP	EMEXT	05B3	2E: FE 06 023B R		LSR3:	ADD	SI,80
0426	4A		DEC	DX	05B8	83 C6 50			INC	DX
0428	2E: FE 06 023C R		INC	CS:[LF]	05C1	42		LSR4:	CMP	DX,EY
0429	83 EE 50		SUB	SI,80	05C1	42			JZ	LSR7
0431	2E: C6 06 023B R 00		MOV	CS:[UF],0	05C1	42			INC	DX
0433	2E: C6 06 023C R 00		MOV	CS:[UF],0	05C2	83 C6 50			ADD	SI,80
0435	81 FD 2770 R		CMP	BP,OFFSET STK	05C5	84 04			TEST	[SI],AL
0437	75 41		JNZ	ULS19	05C7	75 09			JNZ	LSR5
0439	83 FB 05		CMP	BX,5	05C9	2E: C6 06 023C R 00			MOV	CS:[LF],0
0441	72 36		JC	ULSXX	05CF	EB 25 90			JMP	LSR6
0443	26: FF 05		INC	WORD PTR ES:[DI]	05D2	2E: 80 3E 023C R 00		LSR5:	CMP	CS:[LF],0
0445	26: 8B 05		MOV	AX,ES:[DI]	05D8	75 1C			JNZ	LSR6
0447	B9 0006		MOV	CX,6	05DA	2E: 89 4E FE			MOV	CS:[BP-2],CX
0449	F7 E1		MUL	CX	05DA	2E: 89 56 FC			MOV	CS:[BP-4],DX
0451	83		XCHG	AX,BX	05DE	2E: 89 56 FC			SUB	BP,4
0453	26: 89 01		MOV	ES:[DI+BX],AX	05E2	83 ED 04			CMP	BP,OFFSET EMG
0455	8B C8		MOV	CX,AX	05E5	81 FD 0770 R			POP	SI
0457	2E: A1 0241 R		MOV	AX,CS:[XSUM]	05E9	5E			POP	SI
0459	2E: 8B 16 0243 R		MOV	DX,CS:[XSUM]	05EC	5E			POP	SI
0461	F7 F1		DIV	CX	05ED	5E			POP	SI
0463	26: 89 41 02		MOV	ES:[DI+BX+2],AX	05EE	E9 031D R			JMP	EMEXT
0465	2E: A1 0245 R		MOV	AX,CS:[YSUM]	05F1	2E: FE 06 023C R		LSR5X:	INC	CS:[LF]
0467	2E: 8B 16 0247 R		MOV	DX,CS:[YSUM]	05F6	4A		LSR6:	DEC	DX
0469	F7 F1		DIV	CX	05F7	83 EE 50			SUB	SI,80
0471	26: 89 41 04		MOV	ES:[DI+BX+4],AX	05FA	84 04			TEST	[SI],AL
0473	81 3D 03E8		CMP	WORD PTR ES:[DI],1000	05FC	74 1E			JZ	LSR8
0475	74 03		JZ	ULSX	05FE	30 04			XOR	[SI],AL
0477	E9 02CE R		JMP	INIT2	0600	2E: 01 0E 0241 R			ADD	CS:[XSUM],CX
0479	E9 0322 R		JMP	SEEXT	0605	73 05			JNC	ULSR7
0481	2E: 8B 56 00		MOV	DX,CS:[BP]	0607	2E: FF 06 0243 R			INC	CS:[XSUM]
0483	2E: 8B 4E 02		MOV	CX,CS:[BP+2]	060C	2E: 01 16 0245 R		ULSR7:	ADD	CS:[YSUM],DX
0485	83 C5 04		ADD	BP,4	0611	73 05			JNC	ULSR8
0487	E8 0624 R		CALL	?XYADR	0613	2E: FF 06 0247 R			INC	CS:[YSUM]
0489	84 04		TEST	[SI],AL	0618	43		ULSR8:	INC	BX
0491	74 A7		JZ	ULS18X	0619	E9 056E R			JMP	LSRC
0493	52		PUSH	DX	061C	8A C4			MOV	AL,AH
0495	51		PUSH	CX	061E	5E			POP	SI
0497	56		PUSH	SI	061F	59			POP	CX
0499	8A E0		MOV	AH,AL	0620	5A			POP	DX
0501	83 F9 00	LSRC:	CMP	CX,8X	0621	E9 034E R			JMP	ULSRC
0503	75 03		JNZ	LSR0	0624	51		?XYADR:	PUSH	CX
0505	E9 061C R		JMP	LSR8	0625	52			PUSH	DX
0507	D0 E0	LSR0:	DEC	CX	0626	B8 0050			MOV	AX,80
0509	73 03		JNC	AL,1	0629	F7 E2			MUL	DX
0511	B0 01		MOV	AL,1	062B	8B F0			MOV	SI,AX
0513	4E		DEC	SI	062D	8B C1			MOV	AX,CX
0515	83 FA 00	LSR1:	CMP	DX,SY	062F	B9 0008			MOV	CX,8
0517	74 39		JZ	LSR4	0632	F7 F1			DIV	CX
0519	4A		DEC	DX	0634	03 F0			ADD	SI,AX
0521	83 EE 50		SUB	SI,80	0636	8A CA			MOV	CL,DL
0523	84 04		TEST	[SI],AL	0638	80 80			MOV	AL,80H
0525	75 09		JNZ	LSR2	063A	D2 E8			SHR	AL,CL
0527	2E: C6 06 023B R 00		MOV	CS:[UF],0	063C	5A			POP	DX
0529	EB 25 90		JMP	LSR3	063D	59			POP	CX
0531					063E	C3			RET	



＜リスト1＞ つづき

[illegible]



```

0728 DO E8
072A AL,1
072C AND
072E JN2
0730 GLPO2
0732 E5 0A
0734 LOOP
0736 E2 F4
0738 DEC
073A FE CA
073C JN2
073E F5 F0
0740 STC
0742 F9
0744 JMP
0746 EB 0E 90
0748 GLPO2:
0750 MOV
0752 AL,AH
0754 OUT
0756 40H,AL
0758 MOV
075A AL,0EH
075C OUT
075E 46H,AL
0760 INC
0762 AL
0764 OUT
0766 46H,AL
0768 CLC
076A POP
076C GLPOEX:
076E POP
0770 CX
0772 OUT
0774 5A
0776 POP
0778 5A
0780 RET

```

Working

```

0749 0D 0A 1B 54 31 32
074F 1B 49 31 32 36 30
0756 0D 0A 1B 41 0D 0A
075D 0A [ ?? ]

0767 ????
0769 ????
076B ????
076C ????
0770 1000 [ ???? ]

0770
0770

```

ENDS  
START

```

10 '===== ISC-50 応用プログラム ===== 1987 by Digic
11
20 CLEAR,&H3000 : DEF SEG=&H3000 : SCREEN 3 : CLS 3 : BLOAD"MD5"
30 DIM AX(2,1000) : MD5=0 : ANALYZ=3 : GLP=6 : SPARA=9 : NP=10
35 CLS=STRING$(60," ") : COLOR=(1,7)
40 CLS : LOCATE 0,18 : PRINT "***** MENU *****"
50 PRINT:PRINT "1 検索 2 画面コピー 4 ファイル:PRINT
60 PRINT:PRINT "3 画面コピー 5 画面コピー 6 画面コピー
70 AS=INKEY$: IF AS=" " THEN GOTO 100 ELSE IF VAL(AS)=0 THEN END
75 IF VAL(AS)>4 THEN 40 ELSE IF VAL(AS)=0 THEN END
80 AX=VAL(AS) : CLS : ON AX GOSUB 100,200,500,600
90 GOTO 40
91
100 LOCATE 0,18 : PRINT "----- 画面コピー -----":PRINT
110 INPUT "画面コピー (上=0, 下=1):" : AX : PRINT: POKE NP,AX
120 INPUT "画面コピー (2=5, 0=MENU):" : AX
130 IF AX=0 THEN RETURN ELSE IF AX>255 THEN 150
140 POKE SPARA,AX : CALL MD5
150 LOCATE 38,22 : PRINT " : LOCATE 0,22 : GOTO 120
151
200 LOCATE 0,18 : PRINT "----- 画像パターンの解析 (全ドット=63504) -----":PRINT
210 INPUT "パターンの色数 (黒=0, 白=1):" : CX : PRINT
220 AX(0,0)=CX : AX=VARPTR(AX(0,0),1) : BX=VARPTR(AX(0,0)) : CALL ANALYZ(AX,BX)
230 IF AX(0,0)=0 THEN 370 ELSE IF AX(0,0)=1 THEN 380
240 A=0 : FOR I=1 TO AX(0,0)
250 IF AX(0,I)<0 THEN A=A+(65536+AX(0,I)) ELSE A=A+AX(0,I)
260 NEXT I
270 LOCATE 0,22 : PRINT "パターンの分布数は:" : AX(0,0) : " ドット合計は:" : A : "です。"
280 LOCATE 0,22 : PRINT "個々のパターンを調べますか (Yes=1 No=0) ?"
290 AS=INKEY$: IF AS=" " THEN 290 ELSE IF VAL(AS)=0 THEN 350
300 LOCATE 0,22 : PRINT CLS : X=AX(1,1) : Y=AX(2,1)
310 LOCATE 0,22 : INPUT "パターン番号 : AX
320 LOCATE 14,22 : PRINT " : LOCATE 14,22 : PRINT AX
330 IF AX=0 THEN 350 ELSE IF AX>AX(0,0) THEN GOSUB 2000 : GOTO 295
340 GOSUB 2000 : A=AX(0,AX) : IF A<0 THEN A=A+65536
350 LOCATE 24,22 : PRINT "ドット=" : A : " : X=AX(1,AX) : Y=AX(2,AX)
360 GOSUB 1000 : GOTO 300
370 GOSUB 2000 : RETURN
380 PRINT "パターンの分布数が1000以上です。" : GOTO 390
390 AS=INKEY$: IF AS=" " THEN 390
400 RETURN
410
500 LOCATE 0,20 : PRINT "----- 画面コピー -----"
510 CALL GLP : RETURN
520
600 LOCATE 0,19 : PRINT "----- ファイル -----":PRINT
610 INPUT "ファイル番号 (0-9):" : AS : DEF SEG=&H800 : PRINT
620 INPUT "書き込み = 1 読み出し = 0" : AX : DEF SEG=&H800
630 IF AX=0 THEN BLOAD AS ELSE BSAVE AS,0,10079
640 DEF SEG=&H3000 : RETURN
641
1000 '----- Set P.Mark -----
1005 DEF SEG=&H8000
1010 AX=INT(X/8) : NX=X-AX*8 : DX=&H180 : FOR I=1 TO NX+1 : DX=INT(DX/2) : NEXT I
1020 AX=AX+Y*80 : POKE AX,DX : POKE AX+80,DX
1030 IF DX=1 THEN POKE AX+1,&H80 : POKE AX+81,&H80
1040 DEF SEG=&H3000 : RETURN
1041
2000 '----- Reset P.Mark -----
2005 DEF SEG=&H8000
2010 AX=INT(X/8) : Y*80 : POKE AX+1,0 : POKE AX+80,0 : POKE AX+81,0
2020 DEF SEG=&H3000 : RETURN

```







# トランスタ技術

## SPECIAL No. 5

青丸

©CQ出版社 1987

1987年9月1日初版発行

1990年9月10日第6版発行

発行人 神戸一夫

編集人 蒲生良治

発行所 CQ出版株式会社

〒170 東京都豊島区巣鴨1-14-2

電話 03(947)6311~6315

振替 東京0-10665

(定価は表四に表示してあります)

印刷・製本 三晃印刷株式会社



## 実用インターフェース設計法

### ●マイコン活用のためのハードウェア技術入門

本書では、Z80などのマイコンの「入力インターフェース」および「出力インターフェース」の設計法を、多くの設計回路図(全33例)とともに、基本からわかりやすく解説しています。

畔津明仁 著

A 5判 212頁

定価1,440円

送料 250円

## DCモータの制御回路設計

### ●安定に、正確に、効率よくまわす技術

本書は、モータの裸の特性を知るための基本的な実験、安定にまわすための各種回路技術、省電力化のためのPWM制御、サーボ系の安定化技術、マイコンとのインターフェース、位置決め制御などを解説します。

谷腰欣司 著

A 5判 200頁

定価1,550円

送料 260円

## ディジタルIC回路の設計

### ●実験で学ぶTTL, C-MOSの応用テクニック

ディジタル技術を、LS TTL, C-MOSロジックICを使って、実際の実験波形を見ながらやさしく解説しています。

湯山俊夫 著

A 5判 256頁

定価1,650円

送料 260円

## 基礎からの映像信号処理

### ●マイコン画像処理/ハード&ソフトの設計・製作

本書では、映像(ビデオ)信号とその処理回路の基礎を解説したあと、実際の実験用画像処理装置の設計過程をていねいに解説します。

畔津明仁 著

A 5判 202頁

定価1,550円

送料 260円

## 基礎からのメモリ応用

### ●ROM/RAMを使いこなす基本技術

最近、メモリICは高集積化、高速化されていますが、応用のための知識や工夫には共通のものがあります。本書では、基本となる知識や工夫について豊富な実例とともに解説します。

中村和夫 著

A 5判 180頁

定価1,440円

送料 260円

## ステッピング・モータの制御回路設計

### ●実用のための基礎技術とマイコンによる制御技術

本書は、ステッピング・モータを自在に制御するための回路技術、制御ノウハウをわかりやすく解説しています。

真壁國昭 著

A 5判 224頁

定価1,650円

送料 260円

## 6809マイコン・システム設計作法

### ●リアルタイム・モニタ/組み込みコンピュータのための

本書では、MPUに6809を用いた組み込み用ボード・コンピュータを例にとり、周辺用LSIなどについて解説し、後半では、リアルタイム・マルチ・タスク・モニタの構造と、ソース・プログラムの詳細な説明をします。

鶴見恵一 著

A 5判 174頁

定価1,545円

送料 260円



